APPLE Ⅱ
微机及汉字系统

郑 邑 编



同济大学出版社

APPLEⅡ 微机及汉字系统

郑 邑 编

同 济 大 学 出 版 社

内 容 提 要

本书针对目前国内广泛应用的 APPLEⅡ微机，扼要地介绍了其基本用法、APPLE SOFT BASIC 语言、DOS 操作系统、监控系统，并着重介绍了汉字系统的组成、汉卡的使用方法、中文编码及输入方法、中文 APPLE SOFT BASIC 语言特点及中文数据管理程序设计、绘图功能等。

本书附有较多的操作使用和程序设计的实例，可供工矿企业、学校、行政部门等微机用户和大专院校有关师生使用参考。

责任编辑　张平官
封面设计　王肖生

APPLEⅡ 微机及汉字系统

郑 邑 编

同济大学出版社出版

（上海四平路 1239号）

新华书店上海发行所发行

同济大学印刷厂印刷

开本 787×1092 1/16 印张：11.375 字数 285千字

1985年9月第1版　1985年9月第1次印刷

印数1—37,000　科技新书目90—200

统一书号 15335·011　定价：2.05元

前　　言

APPLE Ⅱ PLUS 微机是八位微型计算机的代表机种，由于其性能优良，价格低廉，所以受到广大用户的普遍欢迎。虽然，近年来出现了性能更为优良的其他机种，但是，APPLE Ⅱ 微机目前仍然是广泛应用的机种之一。

随着我国四化建设的飞速发展，近年来国内已引进了大量的 APPLE Ⅱ 微机，该机已在工业、科研、教育等部门发挥了良好作用，在企业管理、人事和技术档案管理、工资管理、成绩管理、情报检索等方面的推广应用，也正在日益增多。无疑，随着微机应用的不断普及，如何更好地利用它，使之在四化建设中发挥更大的作用，已成为广大用户日益关心的问题。但是，由于微机用户的人员结构正在逐步发生变化，用英文或汉语拼音输入输出，将给大多数用户带来困难，因而人们对微机处理汉字信息的要求越来越迫切，为此，不少 APPLE Ⅱ 用户已引进了为该机配制的汉卡。

在APPLE Ⅱ 上实现汉字信息处理的方法有两种，一种是在 APPLE Ⅱ 上配备C-PLUS Ⅱ-A 汉卡或 S.C.C.G-83 超级汉卡，以 APPLE SOFT BASIC 语言处理汉字信息，它构成一个较完整的能处理汉字信息的微机系统；另一种是建立汉字软字库实现汉字信息处理功能。

本书在介绍APPLE Ⅱ 微机基本用法、APPLE SOFT BASIC语言、DOS操作系统和监控系统的同时，着重介绍了其汉字系统的组成、汉卡的使用方法、中文编码及输入方法、中文 APPLE SOFT BASIC 语句特点及中文数据管理程序设计，并在介绍 APPLE Ⅱ 绘图功能的同时，详细叙述利用 APPLE Ⅱ 高分辨率图形显示处理能力设计汉字图形表的方法，以适应微机在报表处理、数据管理、情报检索等方面的推广应用。

本书在叙述内容方法的同时，列举了大量的操作使用和程序设计的实例，可供工矿企业、学校、科研、行政部门等应用参考，对大专院校的有关师生也具有参考价值。

在本书的出版过程中，承蒙上海计算机技术研究所杨载明，顾鼎铭两同志细心审阅，在此表示感谢！

目　　录

前言

第一章　APPLEⅡ PLUS 基本操作方法及汉字系统概述 …… 1

第一节　APPLEⅡ 微机概述 …… 1

一　APPLEⅡ微机基本组成 …… 1

二　APPLEⅡ微机的起动 …… 2

三　屏幕显示 …… 3

四　键盘操作 …… 4

第二节　APPLEⅡ微机的基本操作方法 …… 5

一　APPLE SOFT BASIC 程序结构 …… 5

二　主要操作命令 …… 6

三　监控中可执行的控制字符 …… 8

四　屏幕编辑功能 …… 8

五　软盘机及打印机的使用方法 …… 9

第三节　汉字系统概述 …… 9

第四节　汉字处理系统的硬件配置及系统的支持软件 …… 10

一　系统的硬件配置 …… 10

二　系统的支持软件 …… 11

三　C-PLUSⅡ-A 汉卡的规格及使用注意事项 …… 11

第五节　汉字编码 …… 12

一　中文字母 …… 12

二　辅助字形 …… 12

三　取码规则 …… 13

四　重复键(X键) …… 14

第二章　汉字输入及编辑功能 …… 16

第一节　进入汉字系统 …… 16

一　主机的启动 …… 16

二　进入中文 APPLE SOFT BASIC …… 16

第二节　汉字输入及屏幕显示 …… 16

一　汉字屏幕显示 …… 17

二　输入汉字 …… 17

三　汉字、英文交替输入 …… 19

四　输入出错时的处理 …… 19

第三节　汉字系统的编辑功能和监控中可执行的控制字符 …… 20

一　编辑功能 …… 20

二　监控中可执行的控制字符 …… 20

三　中文字符的 ASCⅡ码 …… 21

第四节　汉字系统与英文系统、监控系统之间的相互转换 …… 21

一　由汉字系统进入监控系统 …… 21

二　由监控系统进入汉字系统 …… 21

三　由汉字系统进入英文系统 …………21
第三章　APPLE SOFT BASIC 语言 …………23
第一节　APPLE SOFT 的一些基本概念 …………23
一　数 …………23
二　变量与数组 …………24
三　算术表达式和标准函数 …………25
四　字符串 …………26
第二节　输出语句（PRINT 语句） …………26
一　执行方式 …………26
二　输出格式 …………27
第三节　提供数据的语句 …………29
一　赋值语句(LET 语句) …………29
二　读数语句(READ)和置数语句(DATA) …………30
三　恢复数据区语句(RESTORE 语句) …………31
四　键盘输入语句(INPUT 语句) …………31
五　单个字符输入语句 (GET) …………32
六　暂停语句(END STOP 语句) …………33
七　注释语句(REM) …………33
第四节　转向语句 …………33
一　无条件转向语句(GOTO 语句) …………33
二　条件转向语句 (IF THEN 语句) …………34
三　选择转向语句(ON GOTO 语句) …………39
第五节　循环语句(FOR NEXT) …………40
一　定义数组维数语句(DIM) …………40
二　循环语句结构以及执行过程 …………40
三　多重循环 …………41
第六节　调用子程序语句 …………43
一　调用子程序语句(GOSUB) …………43
二　多分支调用子程序(ON GOSUB) …………44
第七节　字符串函数语句 …………46
一　字符串长度函数 (LEN) …………46
二　CHR$ 函数 …………47
三　ASC 函数 …………48
四　LEFT$ 函数 …………49
五　RIGHT$ 函数 …………50
六　MID$ 函数 …………51
七　STR$ 函数 …………52
八　VAL 函数 …………53
第八节　中文 APPLE SOFT 的一些特殊功能 …………55
一　移动光标横线的 POKE 语句 …………55
二　CALL 语句 …………56
三　控制打印输出的 POKE 语句 …………60
四　控制显示中文字母的 POKE 语句 …………69
第九节　系统函数及自定义函数 …………69

一　系统函数 ……69
二　自定义函数语句(DEF FN) ……71
第十节　光标控制及屏幕显示格式语句 ……72
一　确定光标位置函数(POS) ……72
二　光标控制语句(HTAB, VTAB) ……73
三　屏幕显示格式语句 ……73
四　设定输出速度语句(SPEED) ……74
第十一节　其它语句 ……74
一　设置输入输出通道的语句 ……74
二　CALL 语句 ……75
三　ONERR GOTO 及 RESUME 语句 ……75
四　CLEAR 语句 ……77
第四章　DOS 操作系统 ……78
第一节　磁盘信息的组织 ……78
一　软盘和软盘驱动器 ……78
二　盘文件类型及其存贮格式 ……78
第二节　DOS 操作系统的引导及盘片的初始化 ……80
一　DOS 操作系统的引导 ……80
二　盘片的格式化 ……80
第三节　DOS 命令的格式及调用 ……81
一　DOS 命令格式 ……81
二　DOS 命令的调用 ……82
三　DOS 命令 ……82
第四节　中文自动启动程序 ……88
一　由英文 BASIC 直接进入中文 BASIC ……89
二　主机启动后直接进入中文 APPLE SOFT BASIC ……89
第五节　正文文件 ……90
一　顺序文件 ……91
二　随机文件 ……101
第六节　文件及软盘的复制 ……107
一　复制 APPLE SOFT 程序文件及二进制文件 ……107
二　复制正文文件 ……107
三　复制软盘 ……111
第七节　程序的连接 ……112
一　不需要传递变量值的程序段间的连接 ……112
二　需要传递变量值的程序段间的连接 ……113
第八节　磁盘管理信息 ……114
一　VTOC 表 ……115
二　索引表 ……115
三　道/区表 ……117
第九节　机器语言文件 ……117
一　有关二进制代码文件的 DOS 命令 ……118
二　磁盘驱动程序(RWTS 子程序) ……118
第五章　监控系统 ……122

第一节　存贮器的组织 ········· 122
一　APPLE Ⅱ 存贮器的组织 ········· 122
二　使用 C-PLUS Ⅱ-A 汉卡时存贮器的组织 ········· 124
第二节　APPLE SOFT 程序使用的存贮器 ········· 125
一　定义变量缓冲区的 APPLE SOFT 语句 ········· 125
二　APPLE SOFT 程序使用的存贮器 ········· 126
三　中文 APPLE SOFT BASIC 程序使用的存序贮器 ········· 126
第三节　监控命令 ········· 127
第四节　反汇编 ········· 131
第六章　超级汉卡 ········· 134
第一节　超级汉卡系统盘 ········· 134
第二节　系统修改 ········· 136
一　超级汉卡系统的启动 ········· 137
二　修改 IDSTAMP 文件 ········· 138
三　22 种字母组的选用 ········· 138
第三节　汉字输入及汉字系统存贮器的组织 ········· 139
一　输入操作 ········· 139
二　编辑功能 ········· 140
三　超级汉卡系统与其它系统间的相互转换 ········· 140
四　超级汉卡系统存贮器的组织 ········· 141
第四节　中文 APPLE SOFT BASIC 语言 ········· 142
一　中文 BASIC 特殊功能语句 ········· 142
二　超级汉卡中扩充的和英文 BASIC 兼容的函数及语句 ········· 143
三　扩充的&语句 ········· 145
第七章　APPLE Ⅱ 的绘图功能 ········· 146
第一节　低分辨率图形 ········· 146
一　低分辨率图形显示格式 ········· 146
二　低分辨率图形显示语句 ········· 146
第二节　高分辨率图形 ········· 150
一　高分辨率图形显示格式 ········· 150
二　高分辨率图形语句 ········· 150
三　绘图实例 ········· 153
四　在图形中配注汉字 ········· 154
五　颜色的限制 ········· 155
第三节　高分辨率图形表 ········· 156
一　绘图向量 ········· 156
二　绘图向量在内存中的表示方式 ········· 158
三　图形表 ········· 158
四　建立图形表 ········· 158
五　图形表的调用 ········· 160
六　汉字图形表 ········· 161
附录Ⅰ　中文字母表 ········· 167
附录Ⅱ　APPLE SOFT BASIC 保留字 ········· 168
附录Ⅲ　ASCⅡ 码表 ········· 169

附录Ⅳ　中文字母的 ASCⅡ 码表……………………………………………………………170
附录Ⅴ　APPLE SOFT BASIC 出错信息…………………………………………………… 171
附录Ⅵ　DOS 错误信息………………………………………………………………………… 172

第一章　APPLEII PLUS基本操作方法及汉字系统概述

第一节　APPLEII 微机概述

一、APPLE Ⅱ 微机基本组成

APPLE Ⅱ 微机由主机、键盘(如图1-1-1所示)、显示器、软盘驱动器及打印机等组成。

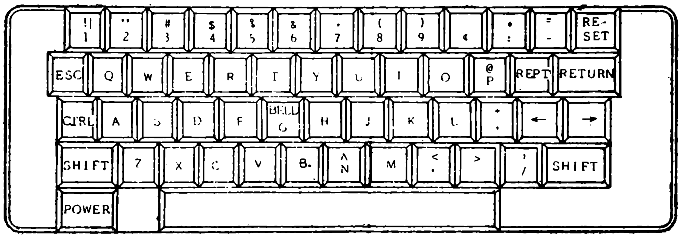


图 1-1-1 APPLE Ⅱ 键盘

1. 主机

CPU：6502；

字长：8 位(bit)；

时钟频率：1MHZ；

指令：56 条，13 种寻址方式。

2. 主存容量

RAM：48K 字节，可扩展到 64K 字节；

ROM：12K 字节。

3. 软盘

尺寸：$5^{1}/_{4}$ 吋；

存储容量：单面单密度约 140K 字节。

4. 显示器

显示格式：24×40 或 24×80；

图形显示能力：低分辨率 48×40

　　　　　　　高分辨率 192×280。

5. 键盘

键数：52 键；

代码：ASC Ⅱ 码。

6. 打印机

可配用 MX-80，MX-100 等。

7. 主机上接口

盒式录音机接口；

键盘输入接口；

视频输出接口；

扬声器输出接口；

游戏接口。

外设扩展插口共八个，编号0～7。除0号外，都可接外设卡，基本配置中打印机控制卡通常插在1号，软盘驱动器控制卡通常插在6号上，16K RAM扩展卡插在0号插座。

另有串行RS-232接口卡，并行IEEE-488接口卡，通讯卡，80列显示卡，语言卡，整数BASIC卡，Z-80卡，ROM编程卡，A/D、D/A转换卡，C-PLUS Ⅱ-A汉卡，S.C.C.G-80超级汉卡等外设控制卡可供选用。

配有DOS3.3操作系统，高级语言有BASIC，FORTRAN，COBOL，PASCAL。加Z-80卡，可使用CP/M操作系统及其支持下的各种语言。

配置汉卡，可直接处理中文信息。

二、APPLE Ⅱ微机的起动

使用APPLE Ⅱ之前，应按系统配置，将外部设备按规定和主机连接。在检查连接正确之后，按下监视器的按钮开关，将亮度控制旋钮调到适当位置，屏幕上将出现光栅，按主机背后的电源开关，可听到响铃声，屏幕显示如图1-1-2所示。

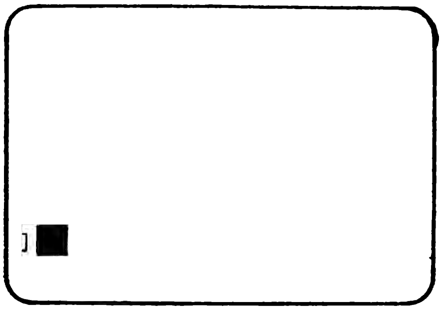


图 1-1-2

当系统主盘（DOS.3.3）放进1号驱动器，打开电源，启动成功则出现如下字幕

```
DOS VERSION 3.3          08/25/80
APPLE Ⅱ PLUS OR ROMCARD SYSTEM MASTER
(LOADING INTEGER INTO LANGUAGE CAR D)
]█
```

]是提示符，表示APPLE Ⅱ当前所处的语言（或工作）方式，闪动的光标表示键盘的输入位置。

系统提示符的显示格式有三种，如表1-1-1。

*：表示APPLE Ⅱ进入监控工作方式，可接受监控命令和输入机器语言程序；

>：表示APPLE Ⅱ进入整数BASIC的工作方式，可以接受整数BASIC的操作命令并可输入程序；

]：表示APPLE Ⅱ进入APPLE SOFT BASIC工作方式，可以接受APPLE SOFT BASIC的操作命令并可输入程序。

提示符格式	语　　言
*	机 器 语 言
>	整　数 BASIC
]	APPLE SOFT BASIC

表 1-1-1

APPLE Ⅱ 监控程序使主机自动进入 APPLE SOFT BASIC 的状态，支持 APPLE SOFT BASIC 的操作系统是DOS3.3，它存放在系统主盘上，当把系统主盘置于 1 号软盘驱动器，开机后，驱动器上的红灯亮，驱动器工作。把 DOS 3.3 装入到内存后进入 APPLE SOFT BASIC 状态。

三、屏幕显示

APPLE Ⅱ 有三种屏幕显示格式：文本显示格式；低分辨率显示格式和高分辨率显示格式。对应每一种显示格式，APPLE Ⅱ 内存 RAM 中各有两个显示缓冲区，分别称为第一页和第二页。

1. 文本显示格式

文本显示格式可在屏幕上显示 960 个字符，即整个屏幕可以显示 24 行，每行 40 个字符如图 1-1-3。显示点阵为 5×7，例如，字符E的点阵如图1-1-4所示。字符间距为横向 2 点，垂直 1 点。

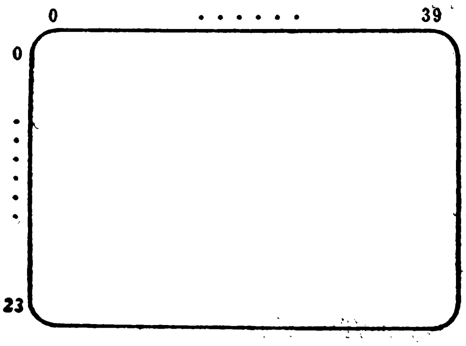


图 1-1-3

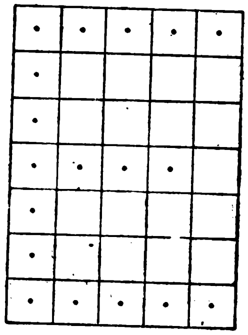


图 1-1-4

正常字符以黑底白字显示，但也可以用白底黑字的反相形式显示，或以闪烁方式显示。

2. 低分辨率图形显示格式

在低分辨率图形显示格式中，APPLE Ⅱ 把屏幕分成 1920 个小方块，即 48 行，每行40 个小方块。如图 1-1-5 所示，每个小方块的颜色有 16 种可供选择，详见第七章。

3. 高分辨率图形显示格式

在高分辨率图形显示格式中，APPLE Ⅱ 把屏幕分成 53760 个小点，即有 192 行，每行 280 个点。如图 1-1-6 所示，每点有六种颜色可供选择。详见第七章。

配备 C-PLUS Ⅱ-A 汉卡时，汉字屏幕使用高分辨率的第二页显示区，当配备 S.C.C.G-Ⅱ 型汉字卡时，可选用高分辨率第一页或第二页显示区显示汉字。汉字的点阵为16×16，实际

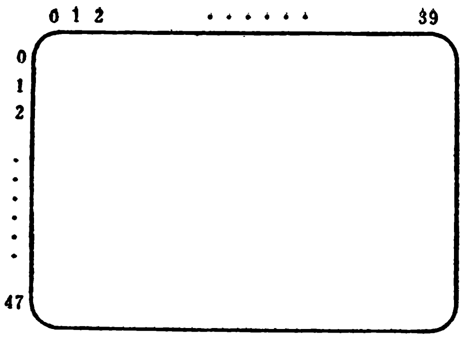


图 1-1-5

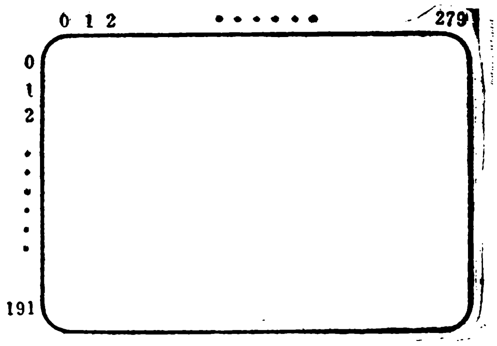


图 1-1-6

使用15×16,例如“同”字的点阵如图 1-1-7 所示。

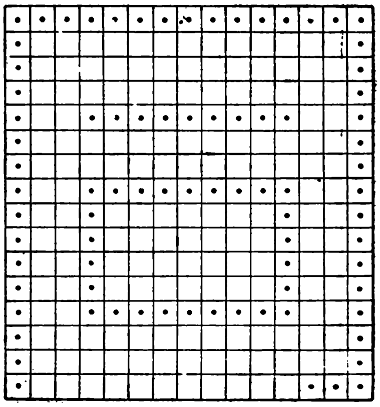


图 1-1-7

四、键盘操作

APPLE Ⅱ 的键盘，除某些键以外，和英文打字机的布局相同，共有 52 个键。

使用的代码是 ASCⅡ 码。

除了 8 个特殊功能键之外，其它按键的字符可直接显示在屏幕上。

8 个特殊键及功能如下：

上下档转换键：SHIFT 2 个；

特殊键：CTRL,ESC,RESET,REPT, ←,→, RETURN。

1．RESET 键

按 RESET 键，中断当前工作，进入复位周期，为防止误操作，RESET 键需和CTRL键同时按(用 CTRL-RESET 表示)才有效。

2．SHIFT 键

SHIFT 键用来输入键帽上部的所示符号，例如当只按 1 键，表示输入数字1，如同时按SHIFT 和 1 键，则表示输入符号“！”，以此类推。但G键上部所示的“BELL”例外，即按SHIFT-G 时 APPLE Ⅱ 不发出声音，而是按 CTRL-G 才发出声音。

3．REPT 键

REPT 键和其它字符键配合操作，可以连续产生出字符，例如按下 A 键，再按 REPT键，A 字符连续出现在屏幕上，直到放开 REPT 键为止。

4. ESC 键

ESC 键必须和其它键配合使用，在 APPLE Ⅱ 监控中，它定义了如下的屏幕编辑功能：

ESC-A，ESC-B，ESC-C，ESC-D 均使光标移动一个字符位置，移动的方向如图 1-1-8 所示。同样，按 ESC 键进入编辑状态后，再按 I、J、K、M 也可使光标移动一个字符位置，移动方向如图 1-1-9 所示。除 I、J、K、M、REPT、CTRL 及 SHIFT 外，按其它任一个键均退出编辑状态。

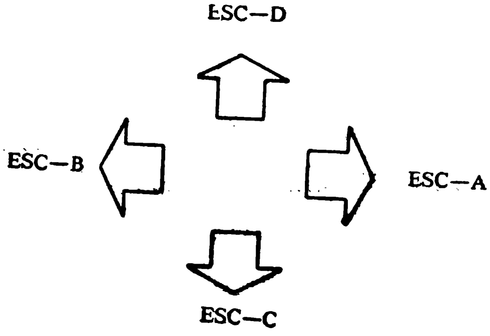


图 1-1-8

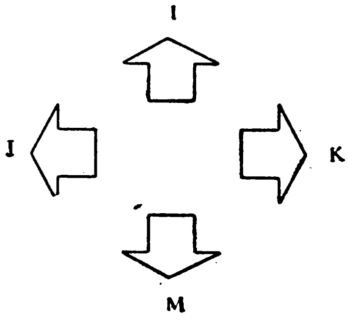


图 1-1-9

ESC 的上述编辑功能只能用于移动光标，当和 REPT 键配合使用时使光标连续移动，存贮器的内容不发生变化。

ESC-E：删除从光标位置到该行结束的所有内容，

ESC-F：删除从光标位置以下的所有屏幕内容，

ESC-@：删除整个屏幕内容。

5. RETURN 键

带有行号的程序语句以 RETURN 键作为输入结束。

从键盘输入命令，按 RETURN 键就执行这个命令。

6. →键

它使光标右移一个字符位置，并把经过的所有字符存入存贮器。

7. ←键

它使光标左移一个字符位置，每次移动时，把经过的字符从程序中清除。

8. CTRL 键

CTRL 键和其它键配合使用时输入特殊的代码，例如按 CTRL-G，输入的 ASC Ⅱ 码为 $07，CTRL 键和其它键配合详见本章下一节叙述。

第二节　APPLEⅡ 微机的基本操作方法

APPLE Ⅱ 提供了一种功能较强的扩展 BASIC 语言——APPLE SOFT BASIC 语言。本节叙述 APPLE SOFT BASIC 程序结构及基本操作方法。

一、APPLE SOFT BASIC 程序结构

下面是一个计算某班级某门课程总分及平均成绩的 APPLE SOFT BASIC 程序。假设 85 分有 10 人，84 分有 5 人，70 分有 15 人，65 分有 8 人。

```
10 LET A=10: LET B=5
20 LET C=15: LET D=8
30 S1=95:S2=84:S3=70:S4=65
40 U=A+B+C+D
50 S=S1*A+S2*B+S3*C+S4*D
60 G=S/U
70 PRINT S
80 PRINT G
90 END
```

由此程序可知，APPLE SOFT BASIC 程序是由若干行组成。并规定，每一程序行中可以包含一个或多个语句。若包含多个语句，每两个语句间要用冒号"："分开，但一行最多只能包含 255 个字符。程序由若干语句集合而成。

一个语句的结构如下：

语句标号　语句定义符　语句体　RETURN

语句标号是 1～63999 之间的一个无符号整数，它识别一行（或一句语句），并规定计算机执行该行的次序。一般机器是按标号大小，由小到大顺序执行。标号可以不按大小顺序写，BASIC 解释程序会作整理并按大小顺序排列，标号可以不连续，以便在修改程序时插入新的语句。

语句定义符指明语句的类型及性质，规定计算机执行某一特定功能，如语句 10 的 LET 表示赋值，语句 70 和 80 的 PRINT 表示打印。

语句体给出需要执行的具体内容，如语句 70 是打印总分 S。通常由常数、变量、运算符等连结而成的表达式组成。

每一程序行以 RETURN 结束。每个程序一般以 END 结束。

二、主要操作命令

操作命令必须以大写英文字母从键盘上输入。

1．NEW 命令

表示清除内存

格式：NEW RETURN

它清除内存中所有程序及变量，通常在输入一个新程序前使用这个命令。当主机启动后，内存中存有 HELLO 程序，须以 NEW 清除后再输入程序。

2．LIST 命令

在屏幕上显示程序清单

格式：(1) LIST RETURN

显示整个程序清单。当一个屏幕页不足以显示整个程序时，程序向上滚动，直到程序结束处。

(2) LIST N RETURN

显示标号为N的语句。例如：LIST 60，显示 60 G=S/U

(3) LIST N1，N2 RETURN

显示从标号 N1 到标号 N2 为止的程序清单。例如：LIST 20，50，显示语句 20，30，

40，50。

(4) LIST，N2 RETURN

显示标号 N2之前的程序清单。例如：LIST，40。显示语句 10，20，30，40

3．DEL 命令

删除程序中的某些语句

格式：(1) DEL N1，N1 RETURN

删除标号为 N1 的语句。例如：DEL 70，70，删除语句70，再用 LIST 命令，语句70不再出现。这个程序不再输出总分。也可以使用 N1 RETURN 删除 N1 语句

(2) DEL N1，N2 RETURN

删除程序中标号为 N1 到标号为 N2 之间的所有语句。

4．RUN 命令

执行一个程序

格式：(1) RUN RETURN

当从键盘上输入程序，计算机并不执行该程序，必须发 RUN 命令，计算机才开始执行该程序，执行到 END 语句止。

(2) RUN N1 RETURN 从程序中行号为 N1 的语句开始执行。

5．CLEAR 命令

清除程序中所有变量。数值变量及数组清除为 0，字符串变量及数组清除为"空字串"。

格式：CLEAR RETURN

6．CONT 命令

继续执行程序

格式：CONT RETURN

对执行 STOP、END 语句后暂停的程序或被以 CTRL-C 键盘操作强迫停止运行的程序重新启动，从断点处继续运行。程序以 STOP，END 语句暂停后，计算机保留各变量的当前值，用户可以执行一些适当的命令，检查和修改有关变量的当前值，也可以检查程序，但不可以改动程序。一旦改动，再用 CONT 语句继续时，系统报错"CANT CONTINUE ERROR"。

7．TRACE 命令

跟踪程序的执行过程

格式：TRACE RETURN

执行这个命令，可以跟踪整个程序执行的全过程。屏幕上连续显示被执行的程序行号。当跟踪到 INPUT 语句，屏幕上出现问号"？"，用户回答后，继续跟踪。便于用户对程序查错。使用这个命令后，程序输出信息和跟踪行号混在一起。通常只在调试程序时使用。

8．NO TRACE 命令

解除跟踪功能

格式：NO TRACE RETURN

解除由 TRACE 命令建立的跟踪功能。

9．HOME 命令

清除整个屏幕显示，但不清除内存 RAM 中的程序。

三、监控中可执行的控制字符

1. CTRL-B　使机器由监控工作方式(*)进入 APPLE SOFT BASIC 工作方式( ])，它破坏现行程序。

2. CTRL-C　当机器处于监控工作方式时，CTRL-C 使机器进入APPLE SOFT工作方式，且不破坏现行程序。

当程序运行时，CTRL-C 中止程序的执行，屏幕上显示信息"BREAK IN N"，指示程序中止在标号为N的语句。输入 CONT 命令，程序继续执行。

3. CTRL-G　使机器发出响铃声。
4. CTRL-J　使光标向下移动一行。
5. CTRL-H　抹去前一个字母。
6. CTRL-M　换行。
7. CTRL-U　复制原屏幕上的字符，相当于→键。
8. CTRL-X　抹去当前一行中的程序语句，并在此语句末尾显示"\"符号。

例如：输入10 PRINT "APPLE SOFT BASIC"后按CTRL-X，删除这个语句，屏幕显示

10 PRINT "APPLE SOFT BASIC"\

9. CTRL-S　用 LIST 命令显示程序清单时，程序语句一行行向上移动，按一下 CTRL-S，停止程序移动，以便阅读程序，再按一下 CTRL-S，程序继续移动。

四、屏幕编辑功能

屏幕编辑功能实现字符的修改，插入和删除。

编辑使用的键：

ESC-A，ESC-B，ESC-C，ESC-D 键；

ESC及I、J、L、M 键；

→，←键；

REPT 键。

编辑时，须使用ESC功能，把光标移到所需行的首字符位置，再修改、插入、删除操作。

1. 修改

例 1　把下面程序中"PRIMT"的"M"修改成"N"

]10 PRIMT "APPLE SOFT BASIC"

]20■

先按 ESC 键，然后按一次 I 键，使光标移到上一行的"PRIMT"的"P"字符，连续按三次 J 键，使光标移到该行的行首的字符"1"，然后用→键使光标右移到"PRIMT" 的 "M"处，键入"N"把"M"修改后，再用→键把光标移到行末，按下 RETURN 键，完成修改操作。

2. 插入

例 2　在下面的程序行中的 "PRINT" 后面插入 TAB(10)

10　PRINT "APPLE SOFT BASIC"

20　■

按例 1 所述的方法把光标移到上一行第一个双撇号的位置，再重新按 ESC 及 I 键使光标移到上一空行，按除I，J、L、M、REPT、CTRL、SHIFT外的任一键，退出编辑状态，再

键入要打入的字符 TAB(10)。重按 ESC 键再按M键,使光标返回语句 10 所在行。并用 J 键把光标移回第一个双撇号处,接着用→键使光标移到行末,按 RETURN 键完成插入操作。

3. 删除

例 3 删除下列"SSOFT"中的第一个"S"。

10 PRINT "APPLE SSOFT BASIC"

20 █

按例 1 所述方法把光标移到上一行的第一个 S 位置,按 ESC 键再按一次 K 键,然后用→键把光标移到行末,按 RETURN 键,完成删除操作。

五、软盘机及打印机的使用方法

软盘机及打印机是微机的主要外部设备。

1. 软盘机的使用方法

从键盘上输入一个程序之后,如果要把一个程序保存到软盘上,以便今后随时调用,可使用如下二条 DOS 命令:

(1) SAVE 文件名

这条命令把存贮器里的程序存到软盘上。文件名由用户自己给出。

(2) LOAD 文件名

这条命令根据指定的文件名把软盘上该程序取到主机的存贮器。如果要执行这个程序,必须再输入 RUN 命令。

软盘在使用之前须经初始化。当 DOS3.3 系统盘放进 1 号软盘驱动器,启动主机后,换上要初始化的软盘,并输入 INIT HELLO 命令,对软盘进行初始化。

其它命令详见第四章叙述。

2. 打印机的使用方法

当打印机的电源接通后,POWER及READY指示灯亮,可以输入命令 PR#1 RETURN 启动打印机。

打印一个程序的清单,输入命令:

PR#1 RETURN

LIST RETURN

打印一个程序的执行结果,输入命令:

PR#1 RETURN

RUN RETURN

打印高分辨率第一页或第二页的图形,分别输入命令:

PR#1 RETURN	PR#1 RETURN
POKE 1913,1 RETURN	POKE 1913, 2 RETURN
CTRL-Q	CTRL-Q

关闭打印机,输入命令:

PR#0 RETURN

第三节 汉字系统概述

汉字数目繁多,形状复杂,汉字如何输入是计算机处理汉字信息时必须要解决的第一个

难题，它涉及汉字编码及汉字的输入方法。汉字的编码方法有音码、形码、音形码等。汉字的输入方法一般可分为整字键盘输入和编码输入，整字键盘输入通常使用的是笔触式大键盘，使用的方法简单易学，但输入速度较慢，硬件费用较高；编码输入一般使用标准的 ASC Ⅱ 键盘，这样就能够和英文系统的键盘兼容。配备汉卡的 APPLE Ⅱ PLUS微机采用中文字母法的编码方法，它是一种形码，这种编码方法选用24个基本中文字母，并选取60个字形作为这些字母的辅助字形，每个汉字可以由 1～5 个中文字母(最多 5 个中文字母)组成，组字的规则大部分均与写字的笔划顺序类同，有效组字数为 23000 个字，中文字母输入是在ASC Ⅱ 键盘上实现的，用 A～W、Y 共 24 个英文字母和 24 个基本中文字母相对应，这样，只要在原来英文系统下使用的 24 个英文字母键上加贴所代表的中文字母就可以很方便地兼容，每个汉字击键仅 1～5 次，操作速度与英文打字机不相上下。

汉字显示及汉字打印是汉字处理系统中的第二难题，有了汉字显示和汉字打印，设计者可以预先在汉字数据处理系统程序中设计汉字提示句子，以便在程序运行时用中文提示用户进行操作。在输出结果时，直接用汉字显示或打印，便于阅读。APPLE Ⅱ PLUS 微机系统能在数据显示终端上以 16×16 的点阵显示汉字，汉字的字幕为 17 字×10行，在汉字输入时，屏幕的左下角会显示“中文”两个小字指示机器处于汉字输入状态，在“中文”两字后面显示每次组字取码时相应的中文字母，使汉字输入过程清晰、明瞭，输入不易出错。APPLE Ⅱ PLUS 通常配备的 MX-80、MX-100 针式打印机以 16×16 的点阵打印输出的汉字，既可以直印，又可以横印，可以输出大字字型，也可以输出小字字型，行与行、列与列的间距均可控制选择，如果配备 MX-80 打印机，每行最多可以打印 60 个汉字，如果配备 MX-100，每行最多可以打印 120 个汉字，可以满足大多数报表打印输出的要求。

汉字处理所遇到的第三个难题是建立汉字库，汉字的库容量依赖于汉字点阵格式。目前，一般用作显示的汉字点阵格式是 16×16，用作打印的点阵格式为 24×24。对于 2 万多个汉字，汉字库的容量是很可观的。微机系统的汉字库可以用硬件建立，其处理速度快，但代价格外高，也可以建在软盘上以降低成本，但处理速度慢。采用三级存放可以提高速度，即一级字库是常用字常驻内存，二级字是缓冲活动库，仅占用少量存储单元，其它的汉字作为三级字库存放在软盘上。APPLE Ⅱ 配备的 C-PLUS 汉卡或超级汉卡是用硬件来构造汉字库，显示和打印的点阵格式都是 16×16。

汉字处理的另一个问题是如何对存储的汉字信息进行加工处理，其核心是如何使中文信息处理和英文信息处理兼容。通常希望在保留原来的英文处理功能基础上，使适用于英文字符处理的一切功能对汉字同样适用，并使软件兼容。APPLE Ⅱ PLUS微机用 APPLE Ⅱ SOFT BASIC 语言调用和处理汉字，如果使用C-PLUS Ⅱ-A 汉卡，除了17 条指令外，英文系统下的 APPLE Ⅱ SOFT BASIC 中的所有 BASIC 指令均保持原有性能，如果使用超级汉卡，则只有 7 条指令不能使用，在汉字处理系统中可以使用 DOS 3.3 操作系统的所有命令。

第四节　汉字处理系统的硬件配置及系统的支持软件

一、系统的硬件配置

APPLE Ⅱ PLUS 主机一台；

监视器一台；
ASC Ⅱ键盘　为操作方便，可以在键盘上 24 个英文字母键的侧面加贴相应的中文字母；
16K RAM 卡一块　插在 0 号扩充插座上；
软盘驱动器控制卡一块　它可以带两个软盘驱动器同时工作，插在 6 号扩充插座上；
软盘驱动器两台　汉字处理系统的程序及数据可以存放在软盘上；
打印机控制卡一块　插在 1 号扩充插座上；
打印机一台　EPSON 公司出品的 MX-80 TYPE Ⅱ或 TYPE Ⅲ 以及 MX-82，MX-100 等均可使用，在打印中文时，要把打印机后面的 4 位拨动开关中的第3位（AUTO FEED）开关设置在 ON 位置；
C-PLUS-A 汉卡或 S.C.C.G-83 超级汉卡一块　插在 1～7 号扩充插座的任何一个插座上，通常插在 3 号扩充插座上。

二、系统的支持软件

自动启动监控程序；
APPLE SOFT BASIC 解释程序；
磁盘操作系统 DOS3.3 版本（当使用 C-PLUS Ⅱ-A 汉卡时）、CCDOS操作系统（当使用 S.C.C.G-83 超级汉卡时）。

三、C-PLUS Ⅱ-A 汉卡的规格及使用注意事项

1．汉卡规格

使用范围：适用于 APPLE Ⅱ个人电脑；
体积：285mm×80mm×15mm；
使用温度，5～35℃；
消耗功率：6W；
中文字幕：17 字×10 行；
英文字幕：34 字×10 行；
中文字矩阵：16×16；
英文字矩阵：8×16；
中文输入方法：中文字母法。

2．电源的规格

+5V：2.5A；
-5V：0.5A；
+12V：1.5A；
-12V：0.5A。
电源的稳定性在 3％以内。

3．汉卡使用注意事项

汉卡带有密封的金属外壳，用户不可随意把外壳拆开以免损坏组件。

安装或取下汉卡时，必须先把电源关闭，并取下主机上方的外壳，小心地把汉卡垂直地插进第 1～7 号扩充插座的任一个上，并仔细地检查安插的位置是否有偏移、不正的情况。打开电源后主机起动情况应和未装汉卡时相同，如果不同，应立即关闭电源，重新安装。

第五节 汉字编码

本节简要介绍中文字母编码法中使用的24个基本中文字母，60个辅助字形以及组字的规则。

一、中文字母

1．中文字母的定义

中文字母编码使用 24 个中文字母，每个中文字母不仅代表一个组字频率高的中文字形，而且都具有完整的意义，其中日、月、金、木、水、火、土、人、心、手、口等字形本身具有完整的意义，直接取作为中文字母。斜(/)，点(.)，交(＋)，叉(×)，纵(丨)，横(一)，钩(亅)等基本笔划和侧(〕)，并(艹)，仰(凵)，纽(女)，方(口)，卜(卜)等基本字形没有完整的意义，分别用和字形相关的具有完整意义的中文字作为中文字母，它们是竹、戈、十、大、中、一、弓及尸、廿、山、女、田、卜。

2．形码

为了能在英文键盘上输入汉字，把 24 个基本中文字母和 A～W、Y 24 个英文字母相对应(英文 X 另外规定为特殊键)，并把与中文字母等价的英文字母称为形码。中文字母与英文形码的对应关系如下：

中文字母	英文形码
日月金木水火土	ABCDEFG
竹戈十大中一弓	H I J KLMN
人心手口	OPQR
尸廿山女田卜	STUVWY
特殊键：	X

3．中文字母组字实例

24 个中文字母可以单独成字，也可以组合起来组字。中文字母编码法规定：一个汉字最多由 5 个中文字母编码组合而成。

例字	中文字母编码	英文形码
日	日	A
伐	人戈	OI
桂	木土土	DGG
哈	口人一口	ROMR
瑚	一土十口月	MGJRB

中文字母不可以再分割，如“王”字，中文字母编码应取“一土”，而不取“一十一”。

二、辅助字形

1．辅助字形规定

仅由24个中文字母不可能把所有的汉字都表示出来。如“四”这个字，尽管很简单，但没有中文字母和它相对应，也不可能由24个基本中文字母组合而成，为此规定了由中文字母演

变出来的60个辅助字形，包括基本中文字母和辅助字形在内的中文字母表见附录I。

2 辅助字形组字实例

辅助字形所代表的仅是符号而不是文字，不能象中文字母那样单独成字，它们可以和中文字母一起组字，也可以由2～5个辅助字形独立成字。

中文字母和辅助字形一起组字实例如下：

例字	中文字母编码	英文形码
欠	弓人	NO
同	月一口	BMR
析	木竹一中	DHML
琼	一土卜口火	MGYRF

辅助字形独立组字实例如下：

例字	中文字母编码	英文形码
四	田金	WC
宏	十大戈	JKI
皑	竹日日山	HAAU
彤	月卜竹竹竹	BYHHH

辅助字形不可单独组字，如单独成字，必须再分解，如“工”字编码取为“一中一”，而不取“一”。

三、取码规则

1. 顺序法

对字形比较简单的汉字，按照字形排列的顺序由外而内，由上而下，由左而右组字。可左右、上下分离的，先左右后上下。

由外而内的是先取外形，再取内形。

例字	外形取码	内形取码	中文字母编码	英文形码
丹	月	卜	月卜	BY
固	田	十口	田十口	WJR

由上而下是先取上形，再取下形。

例字	上形取码	下形取码	中文字母编码	英文形码
内	人	月	人月	OB
巾	中	月	中月	LB
全	人	一土	人一土	OMG

由左而右是先取左形，再取右形。

例字	左形编码	右形编码	中文字母编码	英文形码
仙	人	山	人山	OU
卦	土土	卜	土土卜	GGY
昭	日	尸竹口	日尸竹口	ASHR
徐	竹人	人一木	竹人人一木	HUUMD

2. 字首字身法

凡是字形可以上下、左右分离的字，它的最上方或最左方的字形称为字首。无法上下或

左右分离，但为字典上的惯用字首也看作字首。字首外的全部字形称为字身，字身可上下、左右分离的字，左方或上方字形称为次字首，余下的称为次字身。

字首部分最多取两键。若字形复杂，取首尾两键，中间字形省略。

	例字	字首	字首取码
字首取一码	枝	木	木
字首取二码	召	刀	尸竹
字首取首尾二码	照	昭	日口

字身部分最多取三键。若字形复杂，取首、次、尾三键，中间字形省略。次字首和字首一样最多取首、尾两键。如果次字首可以一码取定的，字身仍取首、次、尾三键。

	例字	字身	字身取码
字身取一码	林	木	木
字身取二码	屏	并	廿廿
字身取三码	行	亍	一一弓
字身取首次尾三码	彬	杉	木竹竹

次字首不能一码取足的，字身取次字首的首、尾及字尾。

例字	字身	次字首	字身取次字首的首尾加字尾
冠	冠	元	一山戈
煎	煎	前	月弓火

3．整体字

字体笔划相连贯无法分离的字称为整体字。

整体字最多取四键，第一键看作为字首，其余看作为字身，如字形复杂，取首、次、三、尾四键。

例字	取法	英文形码
舟	竹月卜戈	HBYI
雨	一中月卜	MLBY

有关中文字母编码的详细叙述请参阅《汉卡中文输入法》一书。①

四、重复键（X 键）

1．重复字的处理

采用中文字母编码法会出现一些重复字。例如

曰与日的编码都为 A

未与宋的编码都为 JD

己与已的编码都为 SU 等。

为了区别重复字，中文字母编码法在两个重复字中的一个字编码的前面加入一个X码。例如

中文字	编码	中文字	编码
日	A	曰	XA
未	JD	宋	XJD
己	SU	已	XSU

①注：《汉卡中文输入法》 佳佳科技事业有限公司编著 松岗电脑图书资料有限公司印行

089581

重复字表见《汉卡中文输入法》一书的附录二。

2．图形字符

在汉字系统中X键与一些字母键组合，可产生不同的图形字符，这些图形字符如同汉字一样输入并出现在程序中。

X键与字母键组合产生的图形字符如图1-5-1所示，图中以XB表示X键与B键的组合，其余以此类推。

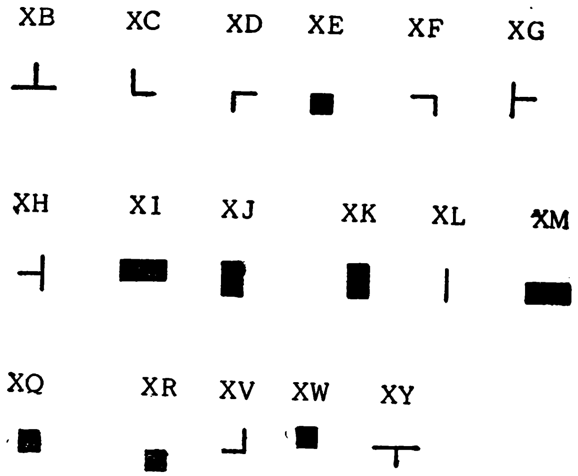


图 1-5-1

第二章　汉字输入及编辑功能

配有汉卡的 APPLE Ⅱ PLUS 处理汉字的程序语言是中文的 APPLE Ⅱ SOFT BASIC 语言。当机器进入中文 APPLE SOFT BASIC，可以输入汉字、编制汉字信息处理程序。

第一节　进入汉字系统

一、主机的启动

装有汉卡的主机启动过程基本上和未装汉卡时相同，如果不使用 DOS 系统，打开电源后机器直接进入英文 APPLE Ⅱ SOFT BASIC，若要使用 DOS 系统，先把 DOS3.3盘放进1号软盘驱动器，打开电源后，DOS 装入到主存的 $9600～$BFFF 区域，并且在屏幕的左下方出现提示符]及闪动的光标，装有汉卡的主机，当把中文自动启动程序作为 HELLO 程序处理时，打开电源后，可以直接进入汉字系统，详见第四章的叙述。装有超级汉卡的主机启动过程及其使用方法将在第六章中叙述。

二、进入中文 APPLE SOFT BASIC

在英文 BASIC 下使用命令 PR#CN，或者把中文自动启动程序加在汉字信息处理用户程序的最前面，都可以进入中文 BASIC，后者将在第四章中叙述。

PR#CN 命令中的 CN 表示汉卡所在的扩充插座号码，如果汉卡插在 3 号扩充插座上，使用 PR#3 命令。

主机启动后，在英文提示符]后第一次输入PR#CN RETURN，屏幕上出现如下字幕①：

```
            ***  佳佳 C-PLUS Ⅱ.A 汉卡  ***
          中文字形产生器发明专利字号：16832
              1983/02/20 改良代号：2.4
                   序号：19512

 ]
```

字幕左下方出现的]是中文 APPLE SOFT BASIC 提示符，它比英文系统中的]大，这就表示已进入汉字系统，用户可以输入中文数据处理程序或调用现有的程序进行中文信息的处理。

此后再次输入 PR#CN 命令，不再出现上述字幕，仅清除整个英文字幕，并在屏幕的左上方显现中文 APPLE SOFT BASIC 的提示符]。

第二节　汉字输入及屏幕显示

使用 C-PLUS Ⅱ-A 汉卡，可以在键盘上交替输入中、英文字母，中英文输入的转换

①注：C-PLUS Ⅱ-A 汉卡以及 S.C.C.G-83超级汉卡均以繁体汉字编码，本书为了方便读者阅读，均以简化字代替。

由 CTRL-L 键控制，当进入汉字系统时，键盘仍设置为英文输入，按了 CTRL-L，键盘转换到汉字输入，若是返回英文输入，再按 CTRL-L 两键。

一、汉字屏幕显示

进入汉字系统，屏幕上可以显示汉字，也可以显示英文字母、数字及其它符号。

在汉字系统中，APPLE Ⅱ使用英文系统中高分辨率图形第二页的存贮区$4000～$5FFF 来显示字幕，这个存贮区能在屏幕上显示 10 行，每行 17 个汉字的字幕。整个屏幕分成两部分，用屏幕的上面10行显示中文 APPLE SOFT BASIC 程序及处理过程，屏幕下面的 1 行固定留给汉字输入时显示汉字输入状态及组字的中文字母。

汉字系统的中文字幕是17字×10行，每个中文字点阵为16×16，英文字幕是34字×10行，每个英文字母、数字或其它符号的点阵为 8×16，即一个中文字占 2 个英文字母的位置，如果一行中显示的是中英文字、数字及其它符号的混合句子，当一行的最右边只剩下一个英文字母的位置而要显示的却是中文字时，系统会自动空下这个位置并转到下一行的起始位置显示中文。

屏幕最下面的 1 行是否显示汉字输入状态及组字的中文字母由 CTRL-D 键控制。如果顺序按下 CTRL-D 及 CTRL-L，屏幕上第 11 行的左方显示“中文”两个小字，提示已进入汉字输入状态。在汉字取码时，“中文”小字的右边显示所取的每个码所代表的中文字母，若要返回英文输入状态，顺序按下 CTRL-L 及CTRL-D，第11行的中文显示随之消失。汉字输入时，如果不先按 CTRL-D，而只按 CTRL-L，屏幕上没有任何改变，即第 11 行不显示“中文”这两个小字。

二、输入汉字

刚进入汉字系统，键盘仍为英文输入状态，用户可以在键盘上输入英文字母、数字及其它字符来编写程序或写入命令。当程序中需要输入汉字时，顺序按CTRL-D及 CTRL-L使键盘转换到汉字输入，屏幕上的第 11 行的左面显示出“中文”两个小字，同时机器发出响铃声，提示用户机器现在开始接受中文输入。用户应该在键盘上顺序输入一个汉字编码的中文字母，每输入一个中文字母都会在“中文”二个小字的右面依次显示这个中文字母。输入一个字的编码后，按空格键，这个中文字就组成了，它显示在光标所指的位置，而左下角原先显示的组字中文字母消失，等待下一个汉字的输入，光标也相应地右移一个汉字位置。

例 1　输入“姓名”两个汉字到字符串变量 A$ 中可以使用语句 INPUT A$，假设执行这个语句时，“？”出现在屏幕的左上角，姓名两个字的汉字编码分别是 VHQM 和 NIR，输入的步骤如下：

按 CTRL-D、CTRL-L，屏幕的左下角出现了“中文”两个小字，见图 2-2-1；

顺序按 V、H、Q、M 键在“中文”两个小字后面出现女、竹、手、一四个中文字母，见图 2-2-2；

按下空格键，第 11 行出现的四个中文字母消失，“姓”字出现在光标处。紧接着光标移到“姓”字的右边，见图 2-2-3；

再顺序按 N、I、R 三个键，第 11 行“中文””两个小字后面出现了弓、戈、口三个中文字母，见图 2-2-4；

再按空格键，第行11显示的弓、戈、口消失，“名”字显示在“姓”字后面。光标移到“名”

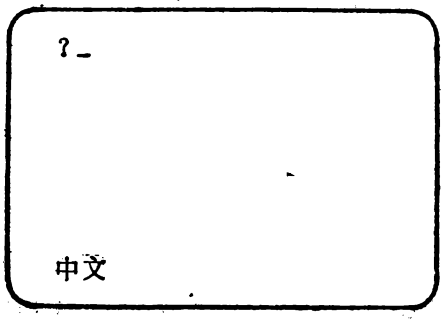


图 2-2-1

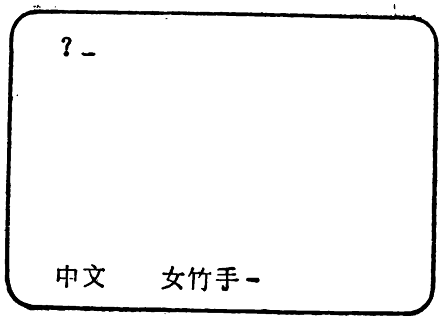


图 2-2-2

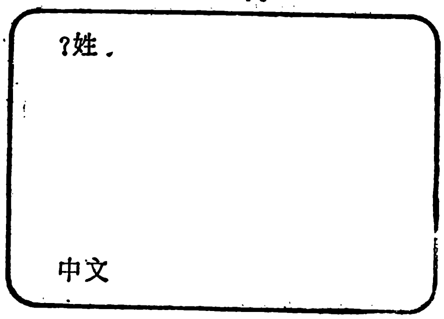


图 2-2-3

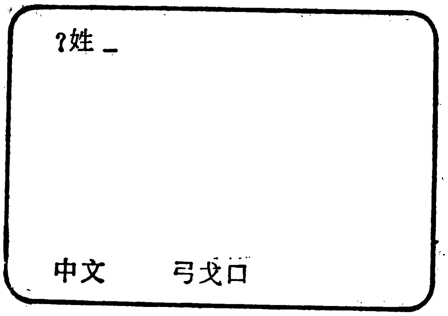


图 2-2-4

字右边，见图 2-2-5。

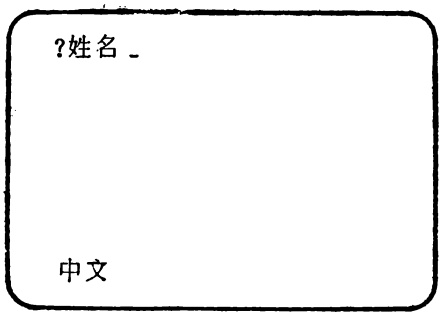


图 2-2-5

汉字输入结束，依次按 CTRL-L，CTRL-D，机器发出响铃声，“中文”两字同时消失，回到英文输入状态，见图 2-2-6。

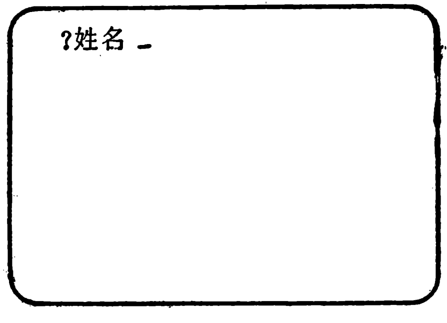


图 2-2-6

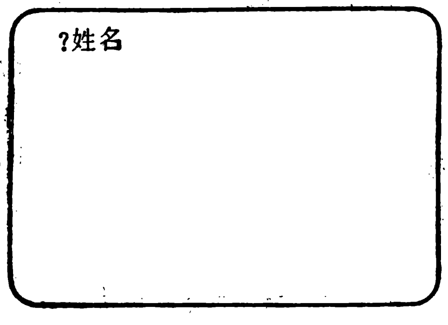


图 2-2-7

最后按 RETURN 键。“姓名”两个字送入字符串变量。如果接下来执行的是 END 语句，提示符出现在下一行起始位置，见图 2-2-7。

RETURN 键仅在英文输入状态下有效，当键盘处于中文输入状态，它不起作用。

三、汉字、英文交替输入

汉字系统中，当键盘处于英文输入状态，可以输入英文字母，数字及其它符号，不允许输入汉字，当处于中文输入状态下，可以输入汉字、数字及其它字符，而不允许输入英文字母。

在输入数据处理程序时，程序语句常出现由中、英文组成的句子，下面举例说明这种语句的输入过程。

例 2　在键盘上输入：

100 INPUT ″请输入姓名：″；A$

在提示符]后输入“100 INPUT″”，接着按 CTRL-D，CTRL-L，进入中文输入状态，按本节第二部分中所述的方法输入“请输入姓名”五个汉字，再按 CTRL-L 和 CTRL-D返回英文输入状态输入余下的符号“：″；A$”，最后按 RETURN 键，语句输入结束。

也可以先在英文输入状态下输入“100 INPUT”，然后转到中文输入状态下输入“″请输入姓名：″；”，再返回英文输入状态输入“A$”。

例 3　在键盘上输入：

200 PRINT ″请输入第″；I；″行″

在英文输入状态下输入“200 PRINT”；

在中文输入状态下输入“　″请输入第″　”；

返回英文输入状态，输入“；I；”；

重新进入中文输入状态，输入“　″行″　”；

最后返回英文输入状态，按 RETURN 键，这个语句输入结束。

也可以按其它的输入顺序输入这个语句，只要记住两点：在中文输入状态下不能输入英文字母；RETURN 键不起作用。

四、输入出错时的处理

1．组字过程中打入了错误的中文字母

例 4　“部门”的“部”，汉字编码是 YRNL，如果打入 YRM 后发现M是错的，处理方法是按一次←键，清除组字区域中M所代表的中文字母，这时，屏幕上有相应的变化即屏幕上第 11 行显示的M所代表的中文字母“一”被清除，当用户重新打入正确的N后，第11 行上原中文字母“一”位置上显示N所代表的中文字母“弓”。再打入 L，最后按空格键，组出“部”字。

2．组成的汉字不是所需要的字

例如：用户欲输入“同”字，其汉字编码是 BMR，但误输入为 BB，而且已经按下空格键，组出了“朋”字，须用←键把光标移动到“朋”字的起始位置，重新输入正确的汉字。用←键移动光标要注意以下事项：

由 1～2 个中文字母组成的字，按一次←键，光标左移一个汉字位置，即按一次←键光标立回到这个字的起始位置；由 3～5 个中文字母组成的字，按一次←键光标仅左移半个汉字

位置，而且返回英文输入状态，屏幕上第 11 行的显示随之消失，再按一次←键，光标才能回到这个字的起始位置，并重新自动返回中文输入状态，重现第 11 行字幕。在按←键时，必须观察光标的移动的位置，否则，有可能破坏上一个汉字，以致不得不从上一个汉字开始重新输入，也可能遗留下原汉字的一半字形，造成输入错误。

3．输入的中文字母不能组成一个汉字或按键超过 5 次

由于不是所有的中文字母的组合都能组成一个汉字，同时，编码规则规定一个汉字最多只能由 5 个中文字母组成，所以当输入的中文字母不能组成一个汉字，例如输入 VFGGM，或按键超过 5 次，机器都会发出响铃声作为错误的警告，这时光标不再移向下一个汉字位置，机器仍然处在中文输入状态，可以直接再次输入正确的中文编码。

第三节　汉字系统的编辑功能和监控中可执行的控制字符

一、编辑功能

英文系统下的 11 个ESC功能在汉字系统中都适用的，它们是：

ESC-A　光标向右移动一个英文字母位置（半个汉字）。

ESC-B　光标向左移动一个英文字母位置（半个汉字）。

ESC-C　光标向下移动一行。

ESC-D　光标向上移动一行。

ESC-E　清除光标横线所在行从横线到行末的整个字幕。

ESC-F　清除光标横线所在位置直到整个屏幕右下角的字幕。

ESCI　进入编辑状态，光标向上移动一行。

ESCJ　进入编辑状态，光标向左移动半个汉字位置。

ESCK　进入编辑状态，光标向右移动半个汉字位置。

ESCM　进入编辑状态，光标向下移动一行。

ESC-@　清除整个屏幕画面。

在使用 C PLUS Ⅱ-A 汉卡的中文 BASIC 中还提供了两句 CALL 语句，其功能等同于 ESC-E，ESC-F，它们可以直接在程序中使用，详见第三章叙述。

二、监控中可执行的控制字符

英文系统中的所有控制字符都保留原有的性能，在汉字系统中另外提供了两个控制字符 CTRL-D 和 CTRL-L。

CTRL-B　由监控进入中文 BASIC。

CTRL-C　不破坏现行程序进入中文 BASIC。

CTRL-G　使机器发出响铃声。

CTRL-J　光标向下移动一行。

CTRL-H　抹去前一个英文字母，当一个汉字由 1～2 码组成，按一次 CTRL-H 就抹去一个汉字，当一个汉字由 3～5 码组成，按一次 CTRL-H 仅抹掉半个汉字。

CTRL-M　换行，相当于 RETURN 键。

CTRL-U　复制原屏幕上的字符，相当于→键。

CTRL-X 抹去现在这一条命令，并在句末显示"\"符号。

CTRL-S 停止或启动屏幕上程序清单的翻动。

CTRL-D 汉字输入时，控制屏幕上第 11 行是否要显示中文输入状态及组字字母。

CTRL-L 控制英文输入与汉字输入之间的切换。

三、中文字母的 ASCⅡ 码

中文字母输入时使用其形码的英文字母键，即中文字母使用小写英文字母的 ASCⅡ 码值。所以，汉字系统中只能使用大写英文字母，比如按 CTRL-L 后，击A键，输入的是中文字母"日"的十进制ASCⅡ码97，存贮在主存中的是其十六进制的 ASCⅡ码61。在英文输入状态下击A键，输入的是英文字母"A"的十进制ASCⅡ码65。中文字母的ASCⅡ码请参阅附录Ⅲ。

第四节 汉字系统与英文系统、监控系统之间的相互转换

一、由汉字系统进入监控系统

在调试程序时，常需要在监控系统下了解程序或数据使用主存的情况。在英文 BASIC 的提示符]后输入 CALL－151 命令可以进入监控系统，屏幕上出现监控系统的提示符*，这里我们称它为英文监控系统提示符。由中文的 BASIC 进入监控系统也使用 CALL－151 命令(－151 的十六进制值是 FF69，是监控程序的入口地址)，但这里出现的监控系统提示符*比英文监控系统的提示符*为大，我们称它为中文监控系统提示符，在这两种提示符下，监控功能相同。

二、由监控系统进入汉字系统

1．在中文监控系统提示符下进入中文 BASIC

使用：

•CTRL-B 这种方法清除原来的中文处理用户程序。

•CTRL-C 这种方法不破坏原有的中文处理用户程序。

2．在英文监控系统提示符下进入中文 BASIC

使用：

•3CTRL-P RETURN 出现中文监控系统提示符*，然后用 CTRL-B 或 CTRL-C 进入中文 BASIC。

也可以先用 CTRL-B 或 CTRL-C 进入英文 BASIC，再用 PR＃3 进入中文 BASIC。

三、由汉字系统进入英文系统

1．由中文 BASIC 进入英文 BASIC

使用：

(1) CALL 64098 命令

地址64098是监控系统中自启动程序的入口地址。执行这个命令后转去执行由地址$FA62(64098 的十六进制数)开始的自启动程序，使机器在不破坏现行程序下进入英文BASIC，屏幕上重现用户上一次使用 PR＃CN 命令时屏幕上的英文字幕，然后紧接着出现一个提示符]

及闪动的光标。由于原来的汉字处理程序没有被破坏，如果用户想再次运行，可以在输入PR＃CN命令后，直接使用RUN命令，如果HELLO程序中有中文自动启动程序及用户程序，也可以用RUN HELLO命令重新进入中文系统，装入并执行用户程序。

（2）CTRL、RESET键

同时按CTRL、RESET两个键也会使机器转去执行自启动程序，进入英文BASIC。

2．在中文监控系统提示符＊下进入英文BASIC

使用：

＊6 CTRL-P　RETURN

假如软盘驱动器中没有放进软盘，那么机器进入英文BASIC。假如软盘驱动器中放进了软盘，进入软盘上面的HELLO程序。

也可以使用CTRL-RESET键进入英文BASIC。

第三章 APPLE SOFT BASIC 语言

APPLE SOFT BASIC 语言是一种功能较强的扩展 BCSIC 语言，具有算术、逻辑和函数运算、字符串处理、直接调用机器语言子程序等功能，并具备比较完善的图形处理能力。

在磁盘操作系统 DOS3.3 的支持下，APPLE SOFT 可处理磁盘文件。

APPLE SOFT 解释程序固化在 APPLE Ⅱ微机的 ROM 中，约占 10K 字节。

配置汉卡的 APPLE Ⅱ微机能对汉字进行处理的程序语言只能是 APPLE SOFT BASIC。汉字是由中文字母组成的。它可以作为字符串处理。凡是可使用字符串变量的 BASIC 语句都可以直接使用汉字。

下述 17 个英文 APPLE SOFT BASIC 语句在配备 C-PLUS Ⅱ-A 汉卡的汉字系统中不能使用。它们是：

VTAB，HTAB，TAB，POS，SPC，TEXT，GR，NORMAL，FLASH，INVERSE，PLOT，HLIN，VLIN，COLOR，SCRN，HOME，HGR。

陈了这 17 个语句外，所有英文 BASIC 语句在汉字系统中基本上都保持了原有性能。

配备 S.C.C.G-83 超级汉卡的中文 BASIC 语言请参阅第六章。

第一节 APPLE SOFT 的一些基本概念

一、数

APPLE SOFT 中的数采用十进制形式。

1．整数

APPLE SOFT 中整数表示范围是 -32767～32767。每个整数在内存中占 2 个字节存贮位置。一个二进制位存放数符，其余十五个二进制位存放数字。

2．实数

APPLE SOFT 中实数表示范围是 -1.70141183×10^{38}～1.70141183×10^{38}。APPLE SOFT 把绝对值小于 2.93873588×10^{-39} 的实数作为 0 处理。

实数输出格式有三种形式：

（1）整数

（2）定点数

有效数字 9 位，超过 9 位，原则上四舍五入。

例如：
```
]PRINT 12.34567896    五入
12.345679
]PRINT 12.34567894    四舍
12.3456789
```

（3）浮点数

当实数的值小于 0.01 或大于 10^8 时，为了阅读方便，以浮点形式显示或打印。

浮点数格式：

±NE±ee

N 正负整数或带有小数的正负实数；

E 表示以 10 为底；

ee 一位或二位整数，是指数值。

例如：]PRINT 1111111114 四舍

1.11111111E+09

]PRINT 1111111115 五入

1.11111115E+09

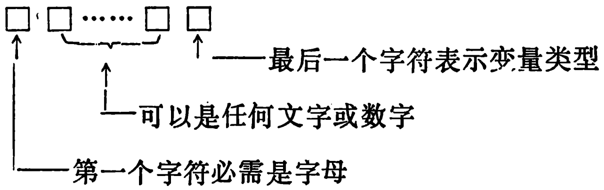
实数在内存中是按二进制浮点格式存放的，占用五个字节存贮位置，一个字节存放指数，另外四个字节存放实数部分。

二、变量与数组

1. 简单变量

能表示不同数值的量叫变量。表示该变量的字符称为变量名。

APPLE SOFT 规定变量名是由字母开头的字母数字串组成，变量名的长度可以是 1～238个字符，最后一个字符指出变量类型。



但 APPLE SOFT 只能识别前面两个字符，例如：

AB 与 ABC 被作为同一个变量。

给变量命名时，应尽量使名字与实际意义相近，以利阅读程序。

APPLE SOFT 中简单变量分为两大类：

（1）数值变量；

（2）字符串变量。

其中数值变量又分为整型变量与实型变量两种。

实型变量名以任意数字或字母结尾，如 A，LN 等；

字符串变量名以"$"结尾，如 A$，UU$ 等；

整型变量名以"%"结尾，如 A%、UU%等。

每个字符串变量存贮的字符个数不能超过 255 个。

不同类型的变量可以以相同的字母数字串(除最后一个 $、%字母)命名。如 A,A$,A%，表示三个不同类型的变量。变量名中不能包含保留字。保留字见附录Ⅱ，它具供命令或程序语句专用的字符串。

2. 数组

APPLE SOFT 中有三类数组，它们是整型数组，实型数组和字符型数组。其命名方式和变量的命名方式相同。数组把一组有序的数或变量作为一个整体来处理。组成数组的变量元素带有相应的下标，称为下标变量。下标个数为 n 的数组称为 n 维数组。

数组表示格式：

一维数组名(下标)　　如：A(I)；

二维数组名(行标，列标)　如：A$(I，J)。

数组 A(I)中若 I=3，则 A(1)，A(2)，A(3)称为下标变量。

APPLE SOFT 中下标下界定为 0。系统初始设定 11 个下标，若超过 11 个，必须用 DIM 语句设定，其最大下标值仅受内存容量的限制。维数最多可达 88 维。数组和简单变量允许取相同的名，但数组名不能相同，即使是不同维数的数组。

三、算术表达式和标准函数

1．算术表达式

算术表达式是由常数、简单变量、下标变量、某些函数等用算术运算符，圆括号组成的式子，其运算结果是一个数值。

APPLE SOFT 中算术运算符及运算的优先级次序如下：

优先级顺序	算术运算符	意义
从高到低↓	∧	乘幂
	−	负号
	*/	乘、除
	+ −	加、减

运算次序：函数优先，其次是乘方，然后是*、/，最后是+、−，括号内优先计算，若括号重叠，则由内向外计算。同一优先级中，依次由左向右。

2．标准函数

APPLE SOFT 中能计算如下一些标准函数：

SIN(X)　　求X的正弦值；

COS(X)　　求X的余弦值；

TAN(X)　　求X的正切值；

ABS(X)　　求X的绝对值；

SGN(X)　　求X的符号函数；

INT(X)　　求X之不大于X的最大整数；

SQR(X)　　求X的平方根(X⩾0)；

EXP(X)　　求X的以 e 为底的指数函数；

LOG(X)　　求X的以 e 为底的对数函数；

RND(X)　　求产生一个[0，1]区间中的均匀分布的随机函数。

说明：

(1) 三角函数中的变量X以弧度给出；

(2) 符号函数定义为

$$SGN(X)=\begin{cases}-1 & X<0\\ 0 & X=0\\ 1 & X>0\end{cases};$$

(3) INT 函数产生一个不大于X的最大整数，例如：

INT(7.6)=7　　　　INT(−5.72)=−6；

（4）自变量可以是一个常数或表达式，且必须用括号括起来，在算术表达式中可直接引用标准函数，例如：

P＝SQR(X)

（5）标准函数可以嵌套使用，例如：B＝SGN(ABS(A))。

四、字符串

日常生活中常遇到大量非数值量，如用英文或拼音字母表示的姓名，它是由字母组成的一串字符，若用汉字表示，则是由中文字母组成的一串字符。字符串是字符的组合，APPLE SOFT 中引进字符串变量来描述。

APPLE SOFT 规定字符串的字符个数应小于 256 个，空格也作为字符计入字符串。汉字当作字符串处理。字符串在程序语句中出现时必须以双撇号括起来，如"ABC"，"中文"等。没有字符的字符串称为空串（用""表示）。在内存中一个字符串中的每个字符占一个字节，这个字符串另外还占用了三个额外字节，其中一个字节存放字符串长度，另外两个字节存放字符串内容的起始字节地址。

几个字符串可以用"＋"号连起来构成新的字符串，称为字符串的连接。

两个字符串可以用关系运算符比较大小。系统还给字符串的处理提供若干个字符串函数，我们将在下面逐步叙述。

第二节　输出语句(PRINT 语句)

APPLE SOFT BASIC 的输出语句(PRINT 语句)可以在屏幕上显示变量或表达式的值以及字符串。如果打印机已被接通，则打印输出。

一、执行方式

1．直接式

PRINT 语句可以在]提示符下直接执行。例如：

```
]PRINT 100*25 RETURN
 2500
]PRINT "ABCD" RETURN
 ABCD
```

PRINT 可以用"?"键代替，要输出的汉字作为字符串出现在语句中，在汉字系统提示符下打入语句：

```
]? "姓名" RETURN
   姓名
```

2．间接方式

当 PRINT 语句出现在程序中，APPLE Ⅱ 把输入的程序存入 RAM，但并不执行程序中任何命令，必须打入 RUN 命令，程序才被执行，当执行到 PRINT 语句时显示或打印输出。

例如：

```
10 PRINT "ABCD"
20 END
```

```
RUN
ABCD
```

二、输出格式

当大批信息输出时，为使结果清晰，人们往往对输出格式有不同的要求，所以 PRINT 语句也有不同的输出格式。

1．基本格式

这是最简单的输出格式，例如：

```
10 PRINT 2*10+5
RUN
25
```

要输出变量的值，必须先对变量赋值

```
10  LET X=10
20  PRINT 3*X
RUN
30
```

一行中可以写上几个 PRINT 语句，例如：

```
10  A$="姓名"：B$="年龄"
20  PRINT A$:PRINT B$
RUN
姓名
年龄
```

PRINT 语句双撇号中的字符全部原样照印，包括空格。

例如：

```
10  PRINT "姓名␣年龄"
20  END
RUN
姓名␣年龄
```

英文系统中，当 PRINT 语句后面没有任何参数时，相当于输出一个空行。在中文 BASIC 中无此功能。

2．固定格式

这种格式在 PRINT 语句中使用逗号分隔输出项，在 APPLE SOFT 中，它把一个输出行分成 3 个区，逗号为输出项在一个区的终止符号，下一个输出项将移到下一个区的起始位置。若输出项在某区显示不下，可在下一个区中继续显示。当第 3 个区用满，则移到下一行的第一个区继续显示。

这三个显示区在屏幕上的位置如下：

```
1                 17                33          40
1_________________1_________________1___________1
    第一个显示区      第二个显示区      第三个显示区
     16个位置          16个位置          8个位置
```

例如：

```
10 A=123:B=456:C=789
20 PRINT A, B, C
RUN
123                456                789
```

当逗号出现在打印语句末尾时，执行该打印语句后不换行，下一个打印语句第一项的值接着打印在本行下一个区的起始位置上。例如：

```
10  LET X=20
20  PRINT "X=", X
```

和

```
10  LET X=20
20  PRINT "X=",
30  PRINT X
```

两程序输出结果相同

```
RUN
X=                20
```

若使用 C-PLUS Ⅱ-A 汉卡，这种格式不能使用。但使用超级汉卡，则可使用这种格式。详见第六章叙述。

3．紧凑格式

这种格式在 PRINT 语句中使用分号分隔输出项，各输出项间不留空格，连续显示在屏幕上。例如：

```
10    A$="姓名":B$="年龄":C$="性别"
20    PRINT A$;"␣␣";B$;"␣␣";C$
RUN
姓名␣␣年龄␣␣性别
```

又如：

```
10  V=540
20  PRINT "总分=";V
RUN
总分=540
```

如果语句最后以分号结束，则输出本语句的信息后不再换行，而紧接着输出下一个 PRINT 语句中的信息。例如：

```
10  V=540:PRINT "总分=";
20  PRINT V
```

结果与上例一样。

4．带有格式函数的 PRINT 语句

在 PRINT 语句中可以使用具有跳过空白位置功能的 SPC 函数及确定各输出项打印或显示的起始位置的 TAB 格式函数。

(1) SPC 函数功能

SPC 函数用在 PRINT 语句中，SPC 后面括号内的数字表示希望跳过空白位置的数目。

例如下面的程序将 TOP SECRET 标题置于屏幕的中央。

```
10  HOME
20  PRINT SPC(15);"TOP SECRET"
30  END
RUN
                    TOP SECRET
```

SPC 后面使用分号。

(2) TAB 函数功能

TAB 功能只能在 PRINT 语句中使用。它指出了打印的位置。例如：

```
10  PRINT "NAME";TAB(10);"STREET";TAB(20);"CITY"
```

它指示从第 10 列开始打印字符串“STREET”，从第 20 列开始打印“CITY”。

TAB 括号中可以是变量或表达式。

若使用 C-PLUS Ⅱ-A 汉卡，这两种函数不能使用。若使用超级汉卡，可用这两种函数。详见第六章叙述。

第三节　提供数据的语句

在汉字系统、英文系统中都可以使用三种提供数据的语句。

一、赋值语句(LET 语句)

上一节已经在程序中使用了赋值语句(LET 语句)，LET 语句把常数或表达式的值赋给变量或把字符串(包括汉字字符串)赋给字符串变量。

例如：求半径 R=10 的圆的面积。

```
10  LET R=10
20  LET S=3.14159*R*R
30  PRINT "面积=";S
40  END
```

LET 语句的执行过程是先计算等号右边的算术表达式的值，并把结果赋给等号左边的变量。如本例中语句 20。

LET 语句中的等号意义与数学上不同，比如 LET X=X+1 在 BASIC 中表示把X原有值加上 1 后再送回X中去，此时X原有的值被X的新值替换了。

LET 语句中的 LET 可省略，要赋值的汉字可以直接作为字符串出现在语句等号右边在双撇号里。例如：

```
10  Y$="同济大学"
```

执行这个语句后，Y$ 中存放了同济大学四个字的编码字母十六进制 ASC Ⅱ 码值。这四个字的汉字编码分别是 BMR，EYX，K，HBND，在字符串变量存贮区的存贮形式是 42 6D 72 20 65 79 78 20 6B 20 68 62 6E 64 20 (中文字母的 ASC Ⅱ 码表参见附录Ⅲ)。使用 C-PLUS Ⅱ-A 汉卡的汉字系统中字符串变量从主存 $9000 开始往下存放。

在字符串赋值语句中，可以把几个字符串用“+”号连接起来构成新的字符串，再赋值给字符串变量。这个字符串的操作过程称为串的联接。例如：

```
10  Y$ = "学生" + "名册"
RUN
学生名册
```

或者用如下的程序：

```
10  A$ = "学生"
20  PRINT A$ + "名册"
```

得到的结果相同。

汉字可以和其它字符组成字符串，例如：

```
10  A$ = "英语"
20  B$ = "80"
30  PRINT  A$ + B$
RUN
英语 80
```

一个数值可以借助于系统提供的字符串函数转换成字符串，再和包含有汉字的字符串相连接。

```
10  A$ = "文件" : X = 1
20  B$ = A$ + STR$(X)
30  PRINT B$
RUN
文件 1
```

STR$ 函数详见本章第七节叙述。

二、读数语句(READ)和置数语句(DATA)

每个 LET 语句只能给一个变量赋值，如果要对多个变量赋值，要使用多个 LET 语句，很不方便。DATA 和 READ 语句配合可以对成批变量或字符串变量赋值。

格式：DATA 数据，数据，……，数据

READ 变量，变量，……，变量

DATA 语句是非执行语句，它为 READ 语句准备数据，READ 语句是可执行语句，执行时从 DATA 语句中顺序取出数据，依次赋给 READ 语句中的各个变量。所以 DATA语句中数据必须按照 READ 语句中变量的先后次序给出，数据类型应和变量类型一致，例如：

```
10  READ A, B, C$
20  DATA 100, 20, "XYZ"
```

程序执行后，A = 100，B = 20，C$ 中为 XYZ。

DATA 语句中的数据个数应大于等于 READ 语句中变量的个数。否则，当数据区的数据用完，而 READ 语句还需要数据，计算机输出出错信息。“OUT OF DATA ERROR”。DATA 语句在原程序中可置于任意位置上。

要输入的汉字可以直接写在 DATA 语句中，以双撇号括起来。例如：

```
10  DATA "姓名", "年龄", "性别"
20  READ A$, B$, C$
```

执行程序后，A$，B$，C$ 中分别存放了“姓名”，“年龄”，“性别”各汉字編码字母的十六

进制 ASC Ⅱ码值。

三、恢复数据区语句 RESTORE 语句

如果要把数据 20，40，字符串“XYZ”，“中文”送到 A，B，C$，D$，E，F，G$，H$及I，J，K$，L$中，用 READ 和 DATA 语句，程序写成：

```
10  READ A, B, C$, D$
20  READ E, F, G$, H$
30  READ I, J, K$, L$
 ⋮          ⋮
100 DATA 20, 40, "XYZ", "中文"
110 DATA 20, 40, "XYZ", "中文"
120 DATA 20, 40, "XYZ", "中文"
```

尽管相同的数据被使用，但程序中要重复写出，因为READ和DATA 语句要配合使用。为了节省程序和内存，DATA 的数据若重复使用一次以上时，可以使用 RESTORE 语句。

APPLE SOFT 中设置一个指针，它指向当前要读的数据是 DATA 语句中的哪一个数据，使用 RESTORE 语句使指针恢复到初始位置，即第一个 DATA 语句的第一个数据前。这样一个 DATA 语句中的数据便可反复使用。

上例可改为：

```
10 READ A, B, C$, D$
20 RESTORE
30 READ E, F, G$, H$
40 RESTORE
50 READ I, J, K$, L$
60 DATA 20, 40, "XYZ", "中文"
```

如果只要重复使用一部分数据，比如上例中若两个 RESTORE 后只要重复使用“XYZ”及“中文”两个数据，并分别送到 E$，F$，G$，H$ 中，必须在程序中作下列处理：虚设一变量X来控制数据的传送。

```
10 READ A, B, C$, D$
20 RESTORE
30 READ X, X
40 READ  E$, F$
50 RESTORE
60 READ X, X, G$, H$
70 DATA 20, 40, "XYZ", "中文"
```

这种处理方法称为“假读”。

四、键盘输入语句(INPUT 语句)

一个变量，如果它的值要由用户在执行程序时加以控制，则可用 INPUT 语句输入。执行 INPUT 语句，系统等待用户从键盘上输入变量值。它是人机对话的重要方式。例如：

```
10 INPUT A, B$
```

```
20  PRINT A, B$
```

执行 INPUT 语句，屏幕上提示"?"符号，输入的数据必须按语句中变量顺序依次输入，各数据之间用逗号分开，数据类型必须和变量类型相同。数据个数和变量个数必须一致。如果输入的数据类型和变量的数据类型不一致，系统在屏幕上显示"REENTER"，并重新执行该 INPUT 语句，用户必须重新输入。如果输入的数据个数小于 INPUT 语句中变量的个数，屏幕上出现"??"，原先输入的数据仍然有效，用户可以继续输入余下的数据。本例中，当执行 INPUT 语句时，用户若在键盘上输入"20, ABC"，则 A=20, B$ 为字符串 ABC。

如果要把汉字输入到字符串变量 A$ 中，使用如下语句：

```
10  INPUT A$
```

执行这个程序时出现"?"，我们可以在"?"后面按照上一章所述的汉字输入方法输入汉字。

在 INPUT 语句中可以写入任意字符串作为程序运行中的提示符。字符串长度不限，只要这个语句不超过 255 个字符即可。语句中最多只能设置一个提示符。例如：

```
100  INPUT "DO YOU WISH TO DO THIS NOW?(Y OR N)";Q$
```

在 INPUT 语句中可以直接写入汉字作提示符，例如：

```
10  INPUT "姓名："; A$
```

执行这个语句时，在屏幕上输出"姓名"两个字，提示用户输入某人的姓名。这时不出现"?"。输入的汉字紧接在"姓名："后显示。

PRINT 和 INPUT 语句配合使用也可以实现上例的功能。

例如：

```
10  PRINT "姓名："
20  INPUT A$
```

不过执行这个程序时，提示符"姓名："出现后，在屏幕下一行显示"?"号。如果希望不出现"?"号，而且输入的数据显示在同一行，使用程序：

```
10  PRINT "姓名：";
20  INPUT "␣";A$
```

设计一个数据处理系统，希望与用户界面简明，方便，易为用户所接受。一般通过系统与用户间的问答方式进行。系统运行时，每一步屏幕上均用中文提示用户应该做什么；出错时用中文指示错误性质，并提示应该如何处理。需要用户作出选择回答时，为了方便用户，对选择项目加以编号，并用中文提示用户怎样用数字代替中文回答，用户只要按相应的数字键，系统就能响应。这样，即使不懂计算机的用户也能很快掌握使用。所以上述 PRINT 和 INPUT 语句配合输出提示符，在编制数据处理系统程序时是很有用的。

五、单个字符输入语句 GET

执行 GET 语句，程序等待从键盘上输入一个键，但不在屏幕上显示这个字符。没有按键，重复执行 GET 语句。

例如：

```
10  PRINT "KEY IN ANY KEY"
20  GET A$
30      ⋮
```

```
40   ⋮
```

程序执行到语句 20 时，等待从键盘上输入任一键后继续执行。

六、暂停语句(END、STOP 语句)

END 语句表示源程序的结束。执行 END 语句，程序中止，屏幕上出现光标。也可以不用 END 语句而终止程序。

STOP 语句暂停程序的执行。执行 STOP 语句时，屏幕上指示暂停在哪一个语句行上，显示信息"BREAK IN N"，N 为标号。暂停时计算机保留各变量的当前值，用户可以输入适当的命令检查程序执行的情况以及采取一些相应的措施，以使程序能正确执行。对于被 END、STOP 语句或 CTRL-C 操作暂停的程序，可以输入 CONT 命令，使程序从暂停处继续执行下去。但在 CONT 命令输入之前，不允许改动程序。

STOP 语句格式：STOP

END 语句格式：END。

七、注释语句 REM

用以对有关程序段注释文字说明。

格式：REM　需注释的内容

这是一个说明性的语句，不被执行。它可以出现在程序的任意位置。常安排在程序段的首行，说明程序段的作用。汉字可以直接出现在 REM 语句中。

第四节　转 向 语 句

程序运行时，通常按标号大小的顺序执行各语句。但是，根据数据处理的要求，有时需要改变这种顺序，跳过几个语句，转到前面或后面的某一语句去继续执行时，就要用到转向语句。

APPLE SOFT BASIC 语言转向语句有三种，它们是无条件转向语句，条件转向语句及选择转向语句。

一、无条件转向语句(GOTO 语句)

GOTO 语句使程序无条件转去执行 GOTO 后面标号所指出的语句。这个标号必须在程序中出现，否则机器报错。

例 1　编制一个统计学校学生总数的程序。输入的是各个班级的学生数。

```
10  T=0
20  INPUT S
30  T=T+S
40  PRINT "TOTAL IS:";T
50  GOTO 20
60  END
```

每输入一个班级学生数，显示当前的累加值，然后无条件转向语句20，继续输入下一个班级的学生数。可以改用中文提示输入及显示结果。

```
5  REM **统计学生总数**
10 T=0
20 INPUT "下一个班级的学生数:"; S
30 T=T+S
40 PRINT "学生数="; T
50 GOTO 20
60 END
```

这个程序必须在打入 PR#3 命令之后输入并执行。

二、条件转向语句(IF THEN 语句)

在有两个分支的程序中条件转向语句可实现分支转移的控制功能。

语句格式：

IF 逻辑表达式(或比较式)THEN 语句标号

IF 逻辑表达式(或比较式)GOTO 语句标号

执行这个语句，先计算逻辑表达式的值，若值为真，则控制转向到 THEN 或 GOTO后面语句标号所指的语句去执行，若为假，继续顺次执行下一个语句。如图 3-4-1 所示。

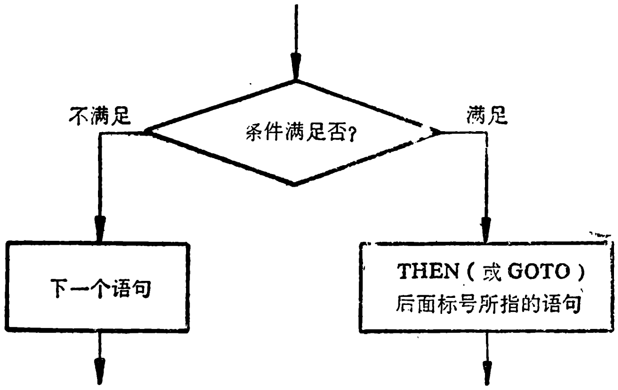


图 3-4-1

1．关系运算符及比较式

APPLE SOFT BASIC 中使用关系运算符如下：

关系运算符	含　义	含义相当的算术符号
=	等于	=
<	小于	<
>	大于	>
<=	小于或等于	≤
>=	大于或等于	≥
<>	不等于	≠

比较式是用关系运算符把两个表达式或两个字符串连接起来的式子。

格式：

算术表达式　关系运算符　算术表达式

字符串　　　关系运算符　字符串

比较式的运算次序是按自左至右且算术运算符优于关系运算符的原则进行的，比较的结果是一个逻辑值，即真与假。例如：

比较式	含　意
B>=5	当 $B \geqslant 5$ 时比较式为真，否则为假
X<>Y	当 $X \neq Y$ 时比较式为真，否则为假
T<B∧2	当 $T<B^2$ 时比较式为真，否则为假
A+B*C>=D-E	当 $A+BC \geqslant D-E$ 时比较式为真，否则为假
A\$<>B\$	当两个字符串 A\$ 和 B\$ 不相等时比较式为真，否则为假
A\$="大学"	当字符串 A\$ 为汉字字符串"大学"时，比较式为真，否则为假

例 2　铁路托运行李规定用每张客票托运行李不超过 50 公斤按运价计算，设每公斤收费0.15元，超过部分每公斤按运价的 2 倍计算。试设计一个程序，输入行李重量、计算托运费并显示。

运费计算公式为：

$$H=\begin{cases}0.15\times Z & (Z\leqslant 50)\\ 50\times 0.15+(Z-50)\times 0.30 & (Z>50)\end{cases}$$

编制程序如下：

```
10  INPUT "重量："; Z
20  IF Z>50 THEN 50
30  H=0.15*Z
40  GOTO 60
50  H=50*0.15+(Z-50)*0.30
60  PRINT "运费=";H; "元"
70  END
```

通常，为便于编写分支问题的程序，常先画出分支程序的流程图。

例 3　有奖储蓄的号码是一个六位整数，最低两位数字为23的奖券是本次开奖的中奖者，编写一个程序，输入各奖券号码及姓名，显示各中奖者的号码、名字，并统计中奖人数。假设以输入一个负的号码作为输入结束标志。流程图如图 3-4-2。

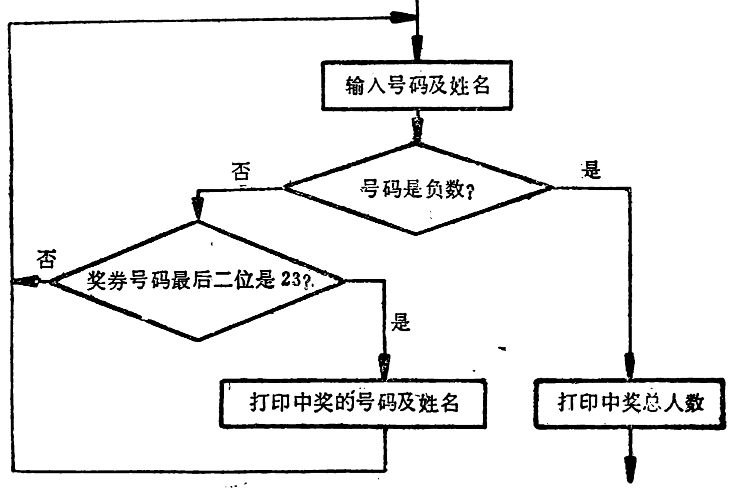


图 3-4-2

程序如下：

```
10  PRINT "号码"; "␣␣␣␣␣␣"; "姓名"
20  READ N, M$
30  IF N<0 THEN 90
40  T=N-INT(N/100)*100
50  IF T<>23 THEN 20
60  I=I+1
70  PRINT N; "␣␣␣␣"; M$
80  GOTO 20
90  PRINT "中奖总人数="; I
100 END
110 DATA 327121, "李明", 256715, "王平"
120 DATA 157625, "王林", 143223, "张红"
130 DATA 456723, "陈力", 216726, "林青"
140 DATA 123758, "王琴", 256723, "张静"
150 DATA -1, "␣"
```

程序中以变量N存放奖券号码，字符串变量 M$ 存放奖券持有者的姓名。

语句 40 可截出奖券号码的最低两位。

例：N=327121

则：T=N-INT(N/100)*100=327121-INT(327121/100)*100

=327121-INT(3271.21)*100=327121-3271*100=327121-327100=21

由于执行语句20，读入N及M$，所以语句150必须以-1及空格作为判终的输入数据，-1赋给N，空格赋给M$。

奖券号码是六位整数，占三个汉字位置。语句10中号码与姓名之间用6个空格分开，接下显示的表格中，号码与姓名之间空两个汉字位置。即

```
号码    姓名
143223  张红
456723  陈力
256723  张静
中奖总人数=3
```

2．逻辑运算符及逻辑表达式

若干比较式以逻辑运算符连接起来组成逻辑表达式，其结果是个逻辑值。APPLE SOFT BASIC 有如下三个逻辑运算符：

逻辑运算符	含义
NOT	非
AND	与
OR	或

（1）NOT　非运算，例如：

NOT A　它表示若A为真，则结果为假，反之亦然；

（2）AND　与运算，例如：

A AND B　它表示 A、B 同时为真时，其结果为真，否则为假；

（3）OR　或运算，例如：

A OR B　它表示 A 为真或 B 为真时，其结果为真，否则为假

其中 A、B 均为比较式，逻辑运算符的优先次序是：NOT，AND，OR，逻辑表达式的运算次序是先括号内，后括号外，并按运算符的优先次序从左至右进行。

逻辑表达式常用于描述复合条件。

例如：

A$ = "大学"AND A>40

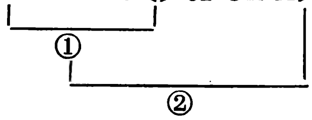
表示A$中存放的是汉字字符串"大学"且 A>40 时，其结果为真。

A> = 90 OR A< = 60

表示 A≥90 或 A≤60 时，结果为真。

A<97 AND A<>32 OR A>121 的运算次序是：

A<97 AND A<>32 OR A>121



表示当 A 小于 97 而不等于 32 或者 A 大于 121 时，结果为真。

例 4　设某班有 5 个学生，要求用 READ-DATA 语句读入学号及数学、物理、英语成绩并输出满足下列条件的学生号码及各门考试成绩，设学号是 4 位整数

（1）三门功课都在 80 分以上。

（2）任一门功课在 95 分以上。

```
10  PRINT "学号"; "␣␣";"英语"; "␣␣";"数学"; "␣␣"; "物理"
20  READ A, B, C, D
25  IF A<0 THEN 80
30  IF B> = 80 AND C> = 80 AND D> = 80 THEN 60
40  IF B> = 95 OR C> = 95 OR D> = 95 THEN 60
50  GOTO 20
60  PRINT A;"␣␣";B;"␣␣␣␣";C;"␣␣␣␣";D
70  GOTO 20
80  END
90  DATA 1232, 68, 75, 82
100 DATA 1234, 85, 87, 92
110 DATA 1236, 65, 98, 72
120 DATA 1238, 78, 67, 93
130 DATA 1240, 81, 92, 97
140 DATA -1, -1, -1, -1
```

语句20一次读入四个变量的值，语句140的四个－1用来作为输入数据结束标志。

由于使用 C-PLUS Ⅱ-A汉卡时，中文 BASIC 中不能使用TAB 功能，要在屏幕上造表格，这里采用语句 10 及语句 60 所示的方法，用空格补位使各栏对齐。在本章第七节中将叙述中文 BASIC 中使游标横线定位的语句，使用这种语句，可以很方便的在屏幕上造表格，

如果使用超级汉卡，详见第六章叙述。

3．字符串及汉字字符串的比较

两个字符串相比较是将它们的字符从左开始逐个进行比较，直到出现不相同的字符。比较这两个不相等字符的 ASCⅡ码，则可确定字符串的大小。若两者含有相等的字符数，而且字符顺序完全相同则两者相等。

例如 "THAT">"THANK"

这是因为前者第四个字符"T"的 ASCⅡ值大于后者"N"的的ASCⅡ值。

又如 "BOOKS">"BOOK"

因为 BOOK 后面是空格，字符"S"的 ASCⅡ值大于空格的 ASCⅡ值。

例 5 用 READ-DATA 语句读入一批国家名称(用英文表示)，编制一个程序显示按英文字母顺序排在最前面的国家名称。

```
5   I=0
10  READ A$
20  IF I=5 THEN 70
30  READ B$
40  IF A$<=B$ THEN 60
50  A$=B$
60  I=I+1:GOTO 20
70  PRINT A$
80  DATA "JAPAN", "ENGLAND", "FRANCE"
90  DATA"CHINA", "GERMANY", "INDIA"
100 END
```

在汉字系统中，汉字字符串的比较是把两个汉字字符串中各个汉字的编码中文字母从左至右逐个比较，直到有不相等的 ASCⅡ值出现，其中 ASCⅡ值较小者则为小的字符串。

例如：用以下的程序比较汉字字符串"高中"与"初中"的大小。

```
10  A$="高中":B$="初中"
20  IF A$>B$ THEN PRINT A$:END
30  PRINT B$:END
RUN
初中
```

因为"高中"的编码为"YRBR"及"L"，"初中"的编码为"YSH"及"L"，当比较到第二个中文字母时，"S"所代表的中文字母"尸"的ASCⅡ值比"R"所代表的中文字母"口"的ASCⅡ值大，所以字符串"初中"大于"高中"。

两个汉字字符串，只有在它们含有相等的中文字母数，而且中文字母顺序相同，才是相等的。否则不等。

例 6 试在教师情况表中，根据文化程度查询并显示其结果，教师情况表由 READ-DATA 语句输入。

```
10  INPUT "文化程度:";U$
20  READ A$, B$, C, D$
30  IF A$="␣" THEN END
```

```
40  IF U$<>D$ THEN 20
50  PRINT A$;"␣␣";B$;"␣␣";C;"␣␣";D$
60  GOTO 20
70  DATA "王林", "男", 46, "大学"
80  DATA "李红", "女", 25, "中专"
90  DATA "陈东", "男", 32, "大专"
100 DATA "张英", "女", 36, "大学"
110 DATA "␣", "␣", 0, "␣"
120 END
```

程序中以空串及0作为教师情况表的结束部分。

如果要查询的对象年龄大于45岁且是大学毕业。程序如下：

```
10  V=45:U$="大学"
20  READ A$, B$, C, D$
30  IF A$="␣" THEN END
40  IF C>V AND D$=U$ THEN 60
50  GOTO 20
60  PRINT A$;"␣␣";B$;"␣␣";C;"␣␣";D$:GOTO 20
70
⋮
⋮
120
```

语句70～120与上例相同。

三、选择转向语句(ON GOTO语句)

这是一种多路分支的转向语句。它根据表达式的值转移到GOTO后面相应的标号所指出的语句继续执行。

格式：ON 表达式 GOTO 标号1，标号2，……，标号K

表达式的数值范围是 0～255，执行这个语句时先对表达式的值取整，若所得的整数为I，则转去执行GOTO后标号I的语句。

例7 统计一个班级学生成绩。学生总人数及每人的成绩以INPUT语句输入。要求统计90～100分、80～89分、70～79分、60～69分及60分以下的学生各有多少人。

下面程序中以S表示学生成绩，利用INT(S/10)+1的函数值把分数分段，A(I)记录各类得分的人数。

```
10  FOR I=1 TO 5
20  A(I)=0:READ  P$(I)
30  NEXT I
40  INPUT "学生总数:"; N
45  PRINT "成绩"
50  FOR I=1 TO N
60  INPUT S
```

```
70  ON INT(S/10)+1 GOTO 80, 80, 80, 80, 80, 80, 90, 100, 110, 120, 120
80  A(1)=A(1)+1:GOTO 130
90  A(2)=A(2)+1:GOTO 130
100 A(3)=A(3)+1:GOTO 130
110 A(4)=A(4)+1:GOTO 130
120 A(5)=A(5)+1
130 NEXT I
140 FOR I=1 TO 5
150 PRINT P$(I);"␣␣";A(I);"人"
160 NEXT I
170 DATA "60分以下","60—69分","70—79分","80—89分","90—100分"
180 END
```

第五节　循环语句(FOR NEXT)

程序设计中常遇到一些要按一定规则重复执行某种操作的问题。使用 FOR NEXT语句可以使程序简洁、明瞭。

一、定义数组维数语句 DIM

APPLE SOFT 系统初始定义数值、字符串数组下标为 0～10，若数组下标超过10必须以 DIM 语句定义数组大小，计算机为数组留出足够的空间。

格式：DIM 一维数组名(下标上界)

　　　DIM 二维数组名(行标上界，列标上界)

这是一个说明性语句，一般出现于程序的首部，或者出现在引用相应数组的语句之前。APPLE SOFT 中数组不允许重复定义，即使是系统初始定义的数组也不能重新定义。

整数数组中每个元素占两个字节，实数数组每个元素占 5 个字节，字符串数组每个元素占 3+N 个字节，其中 N为该元素的字符串长度。数组不宜定义过大，以免超出内存中用户可使用的空间。

程序中数组元素的下标不能超过其定义的下标范围，否则机器报错“BADSUB CRIPT ERROR”。程序中止执行。

二、循环语句结构以及执行过程

循环语句的格式：

FOR X=A TO B STEP C

NEXT X

X 循环控制变量，可用一个简单变量来表示；

A 循环变量初值；

B 循环变量终值；

C 循环变量的增量(或步长)。

A、B、C 可以是数值、变量或表达式。步长可以是正数或负数。如步长为 1，STEP可省

意：FOR 与 NEXT 语句必须成对出现，而且两个语句中的循环变量必须一致。FOR 与 NEXT 语句之间的程序称为循环体，它根据循环变量的变化情况被循环语句重复执行若干次。

循环语句执行过程如图 3-5-1。

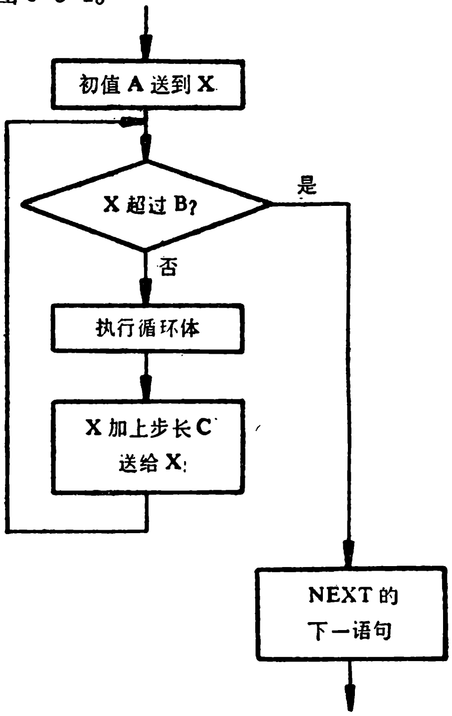


图 3-5-1

例 1　输入一个表格的各栏标题。

```
10  CALL 49941
20  INPUT "总栏数："; N:DIM C$(N)
30  FOR I=1  TO N
40  PRINT "第"; I;"栏标题："；
50  INPUT "";C$(I)
60  NEXT I
```

其中循环变量为 I，初值是 1，终值是N，步长是1，所以语句40，50被执行N次，输入共N栏的栏标题。语句10中的CALL49941清除整个屏幕画面，见本章第八节的叙述。

三、多重循环

实际问题中常出现双重或更多重循环。即循环语句可以嵌套，APPLE SOFT 允许最多 10 层。

图 3-5-2 所示的双重循环中，对 I 的循环语句称为外循环，对 J 的循环语句称为内循环。

多重循环中，只允许从某层循环体内转到该循环体外，而不允许从某层循环体外转入该层循环体内。

例 2　上例中如果要继续输入表格各行的内容，逐行输入时，提示当前输入的是第几行，

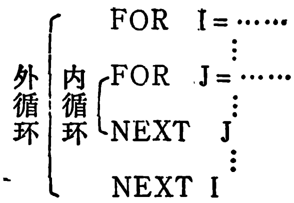


图 3-5-2

并逐栏提示栏标题。程序如下：

```
60  INPUT "总行数："; M
70  DIM D$(M, N)
80  FOR I=1 TO M
90  PRINT "第"; I;"行"
100 FOR J=1 TO N
110 PRINT C$(J);
120 INPUT "";D$(I,J)
130 NEXT J
140 NEXT I
```

语句 100～130 执行内循环，提示一行中每栏的标题并输入各栏标题，语句80～140执行外循环，每输入一行后提示下一行的行号。并继续输入下一行。

若要在屏幕上显示上述表格，程序如下：

```
150 CALL 49941
160 FOR J=1 TO N
170 PRINT C$(J);"␣␣";
180 NEXT J:PRINT ""
190 FOR I=1 TO M
200 FOR J=1 TO N
210 PRINT D$(I, J);"␣␣";
220 NEXT J:PRINT ""
230 NEXT I
240 END
```

例 3　已知一个班级 10 名学生三门课程的成绩，求学生的平均分数，并按名次顺序打印学生的姓名和平均成绩，设这三门课程是数学、物理、英语。姓名存放于A$(I)，平均成绩存放在 L(I)。程序如下：

```
10  FOR I=1 TO 10
20  INPUT "姓名："; A$(I)
30  INPUT "数学："; C
40  INPUT "物理："; D
50  INPUT "英语："; E
60  F=C+D+E
70  L(I)=F/3
80  NEXT I
```

```
90  FOR I=1 TO 9
100 P=L(I)
120 FOR J=I+1 TO 10
130 IF P>L(J) THEN 200
140 P=L(J)
150 L(J)=L(I)
160 L(I)=P
170 U$=A$(I)
180 A$(I)=A$(J)
190 A$(J)=U$
200 NEXT J
210 NEXT I
220 FOR I=1 TO 10
230 PRINT A$(I);"␣␣";L(I)
240 NEXT I
250 END
```

程序中语句90开始的循环，I初始值为1，先用第一个学生的平均成绩L(1)与L(2)比，若L(1)比L(2)小，则将L(1)与L(2)对调，A$(1)与A$(2)对调，然后用新的L(1)与L(3)…，L(10)比，第一次内循环结束。A$(1)及L(1)中放第一名学生的姓名及平均成绩，接着执行第二次循环，I已变为2，这时执行的内循环是把L(3)～L(10)与L(2)比，最后第二名的成绩及姓名调到L(2)及A$(2)中、如此重复，直到全部学生的姓名及平均成绩按名次排好。最后，显示输出。

第六节　调用子程序语句

在一个程序中，有时需要多次进行某种处理，这就必须在程序的不同地方反复写上同一处理程序，这样不仅繁琐，而且容易出错。APPLE SOFT 允许把这段处理程序作为一个子程序供程序的各部分调用，并把调用子程序的程序段称为主程序。

一、调用子程序语句(GOSUB)

调用子程序的语句格式如下：

GOSUB 标号　　标号是子程序的入口

在主程序中用GOSUB 语句调用子程序，在子程序段中执行的最后一个语句是RETURN语句，子程序段中至少要有一个 RETURN 语句。从子程序返回主程序时，返回到本次转子程序语句 GOSUB 的下一个语句并继续执行，如图 3-6-1 所示。

子程序可以嵌套，但不允许递归调用，即不能直接或间接调用子程序本身。

若 GOSUB 指出的标号不存在，机器将报错。

例：

```
10  X=1
20  GOSUB 100
```

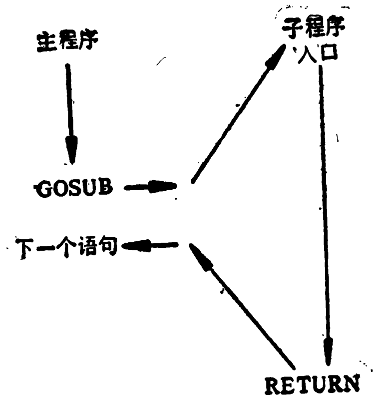


图 3-6-1

```
30  PRINT Y
40  X=5
50  GOSUB 100
60  PRINT Y
70  END
100 Y=(X+10)*5
110 RETURN
```

程序中语句 10～70 为主程序，语句 100～110 为子程序。语句 20 调用子程序，把X=1代入 Y=(X+10)*5，算得 Y=55，然后执行返回语句 110，转而执行语句 30，显示 Y。语句 50 再次调用子程序，算得 Y=75 后返回，执行语句60。

二、多分支调用子程序(ON GOSUB)

ON GOTO 语句可以实现 APPLE SOFT 程序中的多路分支，ON GOSUB 语句可实现子程序分支结构，即转移到子程序序列中的一个或某一部分。

语句格式：

ON　表达式　GOSUB 标号 1，标号 2，……，标号 K

执行此语句时，先对表达式的值取整，若所得整数为 I，则调用 GOSUB 后标号 I 为起始语句的子程序。表达式的规定与 ON GOTO 语句中相同。

一个汉字信息处理系统是直接面向用户的，它与用户界面必须简明，易于操作。一般可采用由系统向用户提示需要操作的命令，而用户只需按下由提示信息所提出的数码即可完成某种选择，有时需要使用多级命令提示完成一种选择。

可以使用 ON GOSUB 语句设计程序。

例 1　一个学生档案管理系统，要求提示第一级命令如下：

0：出口；

1：查询；

2：报表打印。

在查询中要求提供第二级命令：

0：返回；

1：查学生业务档案；

2：查学生健康档案。

在查学生业务档案一项中要求提供第三级命令：

0：返回；

1：按学号查；

2：按课程查；

3：按成绩查。

程序如下：

```
10  CALL  49941
20  PRINT "命令"
30  PRINT "0:出口"
40  PRINT "1:查询":PRINT "2:打印报表":
50  INPUT "选择"; C
60  IF C=0 THEN END
70  ON C GOSUB 200, 300
80  GOTO 10
200 CALL 49941
210 PRINT "命令":PRINT "0:返回"
220 PRINT "1:查学生业务档案"
230 PRINT "2:查学生健康档案"
240 INPUT "选择"; C
250 IF C=0 THEN RETURN
260 ON C GOSUB 400, 500:GOTO 200
     ⋮
300 PRINT "打印报表子程序"
310 RETURN
     ⋮
400 CALL 49941
410 PRINT "命令" :PRINT "0:返回"
420 PRINT "1:按学号查":PRINT "2:按课程查"
430 PTINT "3:按成绩查"
440 INPUT "选择"; C
450 IF C=0 THEN RETURN
460 ON C GOSUB 600, 700, 800:GOTO 400
500 PRINT "查学生健康档案子程序":RETURN
600 PRINT "按学号查找子程序"
610 RETURN
700 PRINT "按课程查找子程序"
710 RETURN
800 PRINT "按成绩查找子程序"
810 RETURN
```

其中语句600，700，800开始的查找程序也是子程序，这是个多级调用的例子，例中CALL 49941语句参阅第八节，它的功能是清除整个屏幕显示。

第七节　字符串函数语句

字符串函数语句可以对字符串进行多种处理。比如求出字符串的长度、取出字符串中的某一部分字符，把数值或ASCⅡ码转换成字符串以及把字符串转换成相应的数值或ASCⅡ码等。由于汉字是由中文字母编码构成的，每一个中文字母都对应一个ASCⅡ码，所以汉字本身可以看作为由中文字母构成的字符串。在汉字系统中，字符串函数语句中字符的概念是对中文字母而言的。一个汉字可以由1～5个中文字母构成，在汉字系统中使用有关字符串的函数语句时要特别注意这一点。

一、字符串长度函数(LEN)

LEN 函数给出一个字符串变量包括的实际字符个数。

例如：

```
10  Q$="APPLE"
20  PRINT LEN(Q$)
30  END
```

函数 LEN(Q$)的值是字符串 Q$ 所包含的字符个数 5。

当用 LEN 函数求一个汉字的长度时，这个长度指的是构成这个汉字的中文字母个数与一个空格符(它组出一个汉字)个数之和。

例如，“部门”的“部”字，中文字母编码为 YRNL，用下面程序可求得其长度为 5。

```
10 LL=LEN("部")
20 PRINT LL
RUN
5
```

如果 Q$ 中存放一个家庭地址“天津路 100/12”，“天津路”的中文字母编码分别为 MK、ELQ、RMHER，下面程序可求其长度：

```
10 Q$="天津路 100/12"
20 L=LEN(Q$)
30 PRINT L
RUN
19
```

其中天津路三个字的长度分别为 3、4、6，后面的数字及斜杠长度之和是 6。

LEN 是函数而不是语句，不能直接作为语句命令使用。例如输入

10 LEN(Q$)语句是错误的。

LEN 函数可以出现在表达式中，因此可以用于赋值语句，条件转向语言，循环语句，打印语句等。

LEN 函数中允许使用字符串表达式，例如：

```
10 A=LEN("APPLE")
```

```
20  B$="1234"
30  C$="CHARACTER"
40  D=A+LEN(B$+C$)
50  PRINT A:PRINT D
RUN
5
18
```

其中B$+C$为字符串表达式，它把B$、C$串接成一个新的字符串"1234CHARACTER"，其长度为13。

二、CHR$ 函数

CHR$ 函数可以将一个 ASCⅡ码转换成相应的字符。

格式：　　CHR$(参数值)

参数值为 0～255。

例如：

```
10  A$=CHR$(48)
20  B$=CHR$(65)
30  PRINT A$;B$;CHR$(49)
RUN
0 A1
```

其中 48，65，49 分别为数字 0，字母 A 及数字 1 的 ASCⅡ码。例如：

```
10 FOR  I=49  TO  57
20 PRINT CHR$(I);
30 NEXT I: PRINT
RUN
123456789
```

在汉字系统中，若参数值是一个中文字母的 ASCⅡ码，当使用 CHR$ 函数时，产生的字符就是这个中文字母，输出到屏幕上时，显示这个中文字母。例如：

```
10  A$=CHR$(97)
20  PRINT A$
30  PRINT"␣"
RUN
日
```

97 是中文字母"日"的 ASCⅡ码，执行这个程序时，显示"日"字。语句30给出了组出汉字的空格符。

用 CHR$ 函数组出一个完整的汉字并显示，可以使用如下程序。设 L 为汉字的编码数(不包括组字的空格符)

```
10  AA$=""
20  INPUT  L
30  FOR I=1 TO L
```

```
40  INPUT A
50  AA$=AA$+CHR$(A)
60  NEXT I
70  PRINT AA$
80  PRINT "␣"
90  END
```

若要组“计”字(“计”的中文字母编码为 YRI，相应的 ASCⅡ码为 121，114，106)，程序执行过程如下：

```
RUN
?3
?121
?114
?106
计
```

先输入编码长度，再分别输入每个中文字母的 ASCⅡ的值，程序执行到语句 80，输出“计”字。

三、ASC 函数

ASC 函数是将一个字符串中第一个字符转换成相应的 ASCⅡ码。例如：

```
10  A=ASC("456")
20  B=ASC(CHR$(48))
30  PRINT A:PRINT B
RUN
52
48
```

其中52是字符串456中第一个字符 4 的ASCⅡ码。而48则是字符 0 的 ASCⅡ码，又如：

```
10  FOR I=ASC("1") TO ASC("9")
20  PRINT CHR$(I);
30  NEXT I:PRINT
RUN
123456789
```

ASC 函数也可以处理汉字字符串。

例如：打印中文字母“日”及相应的 ASCⅡ码和英文形码程序如下：

```
10  A=ASC("日")
20  PRINT "A";"␣␣";"日"; "␣␣";A
RUN
A  日  97
```

例 1　打印一张包含 24 个中文字母及它们相应的 ASCⅡ码、英文形码的表格，程序如下：

```
5  M=24:DIM UU$(M):DIM J(M)
```

```
10  FOR I=1 TO M
20  INPUT UU$(I)
30  J(I)=ASC(UU$(I))-32
40  NEXT I
50  POKE 1403, 1
60  FOR I=1 TO M
70  PRINT CHR$(J(I));"␣␣";UU$(I);"␣␣";ASC(UU$(I))
80  NEXT I
90  POKE 1403 , 0
100 END
```

语句 10～40 的循环语句要求依次输入中文字母“日”～“卜”，中文字母与其相应的英文形码的 ASCⅡ码相差32，例如“日”的 ASCⅡ 码为 97，英文形码 A的ASCⅡ码为 65，语句30求出和中文字母相应的英文形码的 ASCⅡ码。语句 50～90 打印这张表格，它必须在打印机接通之后打出，语句 50 接通打印机，语句 90 关闭打印机，这两个语句将在第八节叙述。

四、LEFT$ 函数

LEFT$ 函数截取字符串中最左边的部分字符。

格式：LEFT$(A$, N)

A$：任一字符串；

N：字符个数。

例如：

```
10  HOME
20  A$="ABCDEF"
30  FOR N=1 TO LEN(A$):PRINT LEFT$(A$, N):NEXT  N
RUN
A
AB
ABC
ABCD
ABCDE
ABCDEF
```

我们可以从某个字符串中取出最左边部分字符再和其它字符串连接。例如：

```
10  A$=LEFT$("ABCD", 2)
20  B$=A$+"CDEF"
30  PRINT A$;B$
RUN
ABABCDEF
```

要从一个包含汉字的字符串中取出一个汉字，必须把这个汉字编码的中文字母连同组字的空格符一起截取出来。

例如，若A$中存放“部门”两个汉字，它们的中文字母编码分别为YRNL和AN，截取中

文字母"卜"(其英文形码为 Y)的程序如下：

```
10  A$="部门"
20  H$=LEFT$(A$, 1)+"␣"
30  PRINT H$
RUN
卜
```

若要把"部"字截取出来并显示，程序如下：

```
10  A$="部门"
20  H$=LEFT$(A$, 5)
30  PRINT H$
RUN
部
```

五、RIGHT$ 函数

格式：RIGHT$(A$, N)

RIGHT$ 函数可以把A$ 中最右边的N个字符截取出来。

例如：

```
10  A$="ABCDEF"
20  PRINT RIGHT$(A$, 2)
RUN
EF
```

又如：

```
10  A$="123456"
20  B$="6789"
30  C$=LEFT$(A$, LEN(A$)-1)+RIGHT$(B$, 4)
40  PRINT C$
RUN
123456789
```

RIGHT$ 函数可以处理含有汉字的字符串。

例如，A$ 中存放"部门"二个汉字，欲截取中文字母"弓"(英文形码为 N)使用如下程序：

```
10  A$="部门"
20  PRINT RIGHT$(A$, 2)
RUN
弓
```

而若要截取"门"字并显示，则程序如下：

```
10  A$="部门"
20  PRINT  RIGHT$(A$, 3)
RUN
门
```

在使用C-PLUS Ⅱ-A 汉卡时，汉字系统下的 LEFT$用缩写L显示、打印。END以E

显示、打印。

六、MID$ 函数

格式：MID$(A$, M, N)

A$：任一字符串；

M：要取出的字符串的起始位置；

N：要取出的字符个数。

字符的位置从最左边算起，最左边字符的位置码为1。

如果N不出现，则从M所指明的位置开始，取出后面所有的字符。

例如：

```
10  PRINT MID$("ABCDE", 3, 1)
20  PRINT MID$("ABCDE", 2, 2)
30  PRINT MID$("ABCDE", 4)
RUN
C
BC
DE
```

MID$ 函数也可以处理包含汉字的字符串。

例如，要从“部门”二个汉字中截取中文字母“口”并显示它的ASCⅡ码，使用如下程序：

```
10  A$="部门"
20  K=ASC(MID$(A$, 2, 1))
30  PRINT K
RUN
114
```

114是中文字母“口”的ASCⅡ码。

在C-PLUSⅡ-A汉卡中，用LEN函数求一个汉字字符串包含的实际字符个数时，求得的值是组成各个汉字的中文字母个数和组出各个汉字的空格键个数之和，而不是这个字符串所包含的汉字个数。但在汉字字符串打印输出时，常常需要求一个汉字字符串的长度，即它由几个汉字组成。若一个字符串由汉字、字母或数字组成，需要求出这个字符串由几个汉字及几个字母、数字组成。由于打印时，一个汉字占据两个英文字打印位置，所以一个包含汉字的字符串长度指的是这个字符串打印输出时占据英文字打印位置的个数。下面给出一个求包含汉字的字符串长度，并把它存放在变量LH中。汉字字符串应预先存放在字符串变量HH$中，中文字母的ASCⅡ码在97～121之间，空格符的ASCⅡ码为32。

程序执行时，按序检查字符串中的每一个字符，如果不是中文字母，则LH加1；如果是中文字母，继续往下查，直到查到一个空格符，表示查到一个汉字，LH加2。

```
110 LH=0:JJ=0
120 JJ=LEN(HH$)
130 IF JJ=0 THEN LH=0:RETURN
140 FOR II=1 TO JJ
150 BB$=MID$(HH$, II, 1)
```

```
160 KK=ASC(BB$)
170 IF KK<97 AND KK<>32 OR KK>121 THEN LH=LH+1:GOTO 190
180 IF KK=32 THEN LH=LH+2
190 NEXT II
200 RETURN
```

在主程序中输入一个包含汉字的字符串，然后转到这个子程序中去执行。如：

```
10  INPUT HH$
20  GOSUB 110
30  PRINT LH
40  END
RUN
? 部门
4
```

七、STR$ 函数

STR$ 函数把一个数值转换成字符串。例如：

```
10  A=1234:B=5678
20  A$=STR$(A):B$=STR$(B)
30  PRINT A$;B$
RUN
12345678
```

语句 20 把数值 A 和 B 转换成字符串。所以 A$="1234"，B$="4567"。

汉字系统中允许使用中文作为文件名，如果文件是按部门组织，要求根据输入的部门编号给各部门的文件自动编码，可以使用如下程序：

```
10  INPUT "文件名："; F$
20  INPUT "部门号："; X
30  F$=F$+STR$(X)
40  PRINT F$
50  END
```

如果输入文件名为"部门"，输入第一个部门的部门编号 1，则对应第一个部门的文件，其文件名为"部门1"。

例 2，输入一串汉字，把其编码值作为一个字符串打印(每个中文字母的 ASCⅡ码值都取 3 位)程序及运行实例如下：

```
10  A$=""
20  INPUT UU$
30  LL=LEN(UU$)
40  FOR I=1 TO LL
50  C$=MID$(UU$, I, 1)
60  J=ASC(C$)
70  IF J=32 OR J=97 OR J=98 OR J=99 THEN 130
```

```
80  A$=A$+STR$(J)
90  NEXT I
100 POKE 1403, 1:PRINT A$
110 POKE 1403, 0
120 END
130 B$="0":B$=B$+STR$(J)
140 A$=A$+B$:GOTO 90
]RUN
? 同济大学
098109114032101121120032107032104098110100032
```

语句20把"同济大学"四个字输入到UU$中，语句130，140把每个中文字母所对应的ASCⅡ码值变为3位数，语句80把多个中文字母的ASCⅡ码串成一个字符串，语句100打印这个字符串。

八、VAL函数

VAL函数把一个字符串转变为数值。

格式： VAL("字符串")

VAL能转换0～9，+，-，·等字符，但组合它们时，必须为合法的数值形式。例如：

```
10  A=VAL("+123")
20  PRINT A
RUN
123
```

VAL函数有下列几个规定：

（1）字符串前面的空白不计；

（2）非数值字符不计；

（3）第一个字符非空白，非数值时，其值为0。

例如：

```
10  A=VAL("00123")
20  B=VAL("R456")
30  C=VAL("456A")
40  D=VAL("P")
50  E=VAL("+789")
60  PRINT A;B;C;D;E
RUN
12304560789
```

例3 设考生外语成绩一栏是由一个汉字和三个数字组成，一个汉字表示语种，三个数字表示分数，如"英080"，表示该考生英语成绩为80分。

设考生考试的语种输入字符串变量A1$，分数输入字符串变量A2$，语种和分数串接起来的字符串保存在A$中，输入程序如下：

```
10  INPUT "考生总数：", M
```

```
20  DIM A$(M):DIM A1$(M):DIM A2$(M)
30  FOR I = 1 TO M
40  INPUT "语种："; A1$(I)
50  INPUT "分数:"; A2$(I)
60  IF LEN(A2$(I)) = 1 THEN A2$(I) = "00" + A2$(I):GOTO 80
70  IF LEN(A2$(I)) = 2 THEN A2$(I) = "0" + A2$(I)
80  A$(I) = A1$(I) + A2$(I)
90  NEXT I
```

其中语句 60，70 把输入的分数拼成三位。如果要统计某一语种，分数在某一分数线以上的人数，程序如下：

```
100 INPUT "语种："; B$
110 INPUT "分数下限："; L
120 FOR I = 1 TO M
130 C$ = "":LL = LEN(A$(I))
140 FOR II = 1 TO LL - 3
150 C$ = C$ + MID$(A$(I), II, 1):NEXT  II
160 IF  C$ = B$ AND VAL(RIGHT$(A$(1), 3))> = L   THEN J = J + 1
170 NEXT I
180 PRINT  B$;L;"分以上共计"; J;"人"
190 END
```

由于分数固定为 3 位数，语句 140～150 取出代表语种的一个汉字，语句160中截取分数并与分数下限比较。

若分数不再规定为 3 位数，例如“英 5”表示英语成绩是 5 分，“日 80”表示日语成绩为 80 分，“英 100”表示英语成绩为 100 分，则程序设计如下：

```
10  INPUT "考生总数："; M
20  DIM A$(M)
30  FOR I = 1 TO M
40  INPUT "语种,分数："; A$(I)
50  NEXT I
100 INPUT "语种："; B$
110 INPUT "分数下限："; L
120 FOR I = 1 TO M
130 C$ = "":LH = 0:LL = LEN(A$(I))
140 FOR II = 1 TO LL
150 K = ASC(MID$(A$(I), II, 1))
160 C$ = C$ + MID$(A$(I), II, 1):LH = LH + 1
170 IF K = 32 THEN
180 NEXT II
190 IF VAL(RIGHT$(A$(I), LL - LH))> = L AND C$ = B$ THEN J = J + 1
200 NEXT I
```

```
210 PRINT  B$;L;"分以上共计"; J;"人"
220 END
```

第八节　中文　APPLE SOFT 的一些特殊功能

在 C-PLUS Ⅱ-A 汉卡系统中，中文 APPLE SOFT 提供下述一些特殊功能的 POKE 及 CALL 语句，用于控制光标横线的移动、屏幕编辑及控制打印机的输出格式等。POKE 及 CALL 语句功能详见第九节及第十一节。

一、移动光标横线的 POKE 语句

汉字系统中用两个 POKE 语句取代英文语言中移动光标横线位置的 HTAB、VTAB 功能，光标横线移动是以一个英文字母为基本单位(即半个中文字)。HTAB、VTAB 语句见第十节。

1. POKE 214，H

这个语句相当于英文 BASIC 中的 HTAB 功能，H 值是光标横线要移动的水平座标值。屏幕最左面的座标值定义为 0，最右面定义为 33。

2. POKE 215，V

这个语句相当于英文 BASIC 中的 VTAB 功能，V 值是光标横线要移动的垂直座标值。屏幕最上方的座标值定义为 0，最下方定义为 9，见图 3-8-1 所示。

如图 3-8-1 所示，光标横线在水平方向是以一个英文字母位置为单位左右移动的。屏幕上一行最多可以显示 17 个汉字，如果要把一个汉字显示在左边第二个汉字位置，H值应为2，光标在垂直方向是以一行为单位上下移动的。利用这两个语句可以把光标横线移动到屏幕上的任何位置。也就是说，可以把汉字显示在屏幕上的任何一个位置上。

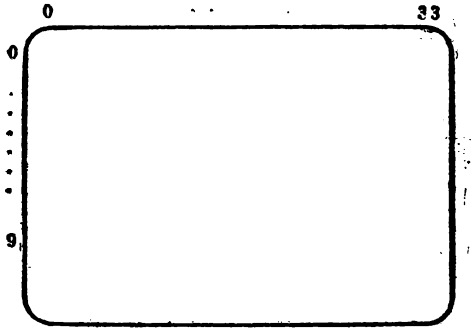


图 3-8-1

例 1　设计一个字幕，要求在屏幕正中间显示如图 3-8-2 所示的字幕。

```
200 CALL 49941
210 POKE 214, 7:POKE 215, 3
220 PRINT "微机汉字报表处理程序"
230 POKE 214, 10:POKE 215, 5
240 PRINT "同济大学计算中心"
250 POKE 214, 13: POKE 215, 7
```

```
260 PRINT "1984/3/27"
270 END
```

语句220在第4行(行编号为3)显示10个汉字，要使它们显示在正中间，两边共需留下7个汉字位置，即左右两边各留3个半汉字的位置，即7个英文字母的位置，所以H值取为7，所以语句210中使用了POKE 214，7。

这个程序可以显示所要求的字幕，但在屏幕第5行，第7行，第9行的左方各留下一个光标横线，为使画面美观、整洁，必须清除这些光标横线。详见下面的CALL语句叙述。

微机汉字报表处理程序

同济大学计算中心

1984/3/27

图 3-8-2

二、CALL语句

设计一个汉字信息处理程序，当程序运行时，都希望有一个整洁、美观的屏幕显示，使整个处理过程清晰，结果一目了然，用户使用这个程序进行操作时也不容易出错，这就要求在程序执行过程中随时清除不必要的显示信息。

APPLE Ⅱ PLUS 的汉字系统提供5个子程序，供用户在程序中使用，以美化屏幕显示。

这5个子程序的调用方法如下：

```
CALL HM
CALL CC
CALL CE
CALL CL
CALL CF
```

各个子程序的入口地址与C-PLUS Ⅱ-A 汉卡所在的插座号码值CN有关。计算公式如下：

设 BS = 49152 + 256*CN 则

HM = BS + 21

CC = BS + 24

CE = BS + 36

CL = BS + 39

CF = BS + 42

如果汉卡插在3号插座上，那末调用语句是

```
CALL  HM      CALL 49941
CALL  CC      CALL 49944
CALL  CE      CALL 49956
CALL  CL      CALL 49959
CALL  CF      CALL 49962
```

下面分述各个子程序的功能：

1. CALL HM

这个子程序可取代英文 BASIC 中 HOME 的功能，执行这个语句，清除整个屏幕画面，并将光标横线移到左上角。如本节例 1 中第一个语句就是 CALL HM 语句，它清除以前遗留在屏幕上的全部内容。

2．CALL CC

执行这个子程序，清除这个命令执行前 PRINT 语句所遗留下来的光标横线。

例 2　在例 1 中使用 CALL CC 语句，可以清除各个 PRINT 语句遗留下来的光标横线。程序修改如下：

```
200 CALL 49941
210 POKE 214，7:POKE 215，3
220 PRINT "微机汉字报表处理程序"
230 CALL 49944:POKE 214，10:POKE 215，5
240 PRINT "同济大学计算中心"
250 CALL 49944:POKE 214，13:POKE 215，7
260 PRINT "1984/3/27"
270 CALL 49944
280 END
```

语句 230，250，270 中都使用了 CALL CC，分别将语句 220，240，260 中 PRINT 遗留下来的光标横线清除。

3．CALL CE

这个子程序的功能和 APPLE 中的英文指令 ESC E 相同，它清除光标横线所在行从横线位置开始到行末的整个字幕。

例 3　设计一个输入报表各栏标题，各栏设定长度及各栏的数据类型（设数据类型分为文字型和数字型两种，分别以 0，1 表示）的程序，要求在屏幕上以表格形式显示输入的数据，假设要求先在第 4 行显示这个表格的表目“栏长度”、“类型”、“栏标题”，当输入各栏长度、类型及标题时从第 5 行开始依序显示在相应的表目下面。如图 3-8-3 所示。

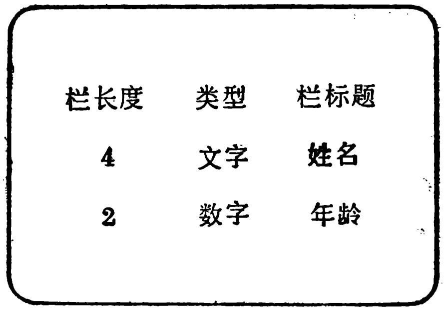


图 3-8-3

下面，先给出显示表目的程序：

```
200 CALL 49941
210 POKE 214，3:POKE 215，3
220 PRINT "栏长度"
230 POKE 214，11:POKE 215，3
```

```
240 PRINT "类型"
250 POKE 214, 17:POKE 215, 3
260 PRINT "栏标题"
270 CALL 49944
```

如果要求在各栏长度、类型及标题输入过程中，每次都在屏幕的第一行第三列开始用中文"请输入栏长度"、"请输入类型"、"请输入栏标题"显示来提示用户操作，当用户输入一个数据后，再填到下面表格的相应位置。程序可以这样设计：先在屏幕第一行第三列显示"请输入"三个字，以后每次都保留这三个字，并接在它后面显示"栏长度"、"类型"、"栏标题"以构成一个完整的提示信息，每当一个数据输入后，把光标横线移到第一行第四个字的起始位置，用 CALL CE 语句把"请输入"后面的整行字幕清除，以显示下一个提示信息。现在接着上面的程序往下写，假设只输入一行。

```
275 FOR K=1 TO 5:POKE 214, 0:PRINT K:NEXT K
280 POKE 215, 0:POKE 214, 2
290 PRINT "请输入";
300 INPUT "栏长度:"; LN
310 POKE 214, 5:POKE 215, 4:PRINT LN:GOSUB 800
320 INPUT "类型 0:文字 1:数字"; LM
330 IF LM=1 THEN 370
340 IF LM=0 THEN 360
350 GOTO 320
360 POKE 214, 11:POKE 215, 4:PRINT "文字":GOSUB 800:GOTO 380
370 POKE 214, 11:POKE 215, 4:PRINT "数字":GOSUB 800
380 INPUT "栏标题";C$
390 POKE 214, 17:POKE 215, 4:PRINT C$:GOSUB 800
400 END
    ⋮
800 POKE 215, 0:POKE 214, 8
810 CALL 49956
820 RETURN
```

程序中语句800开始的子程序把屏幕第一行中中文提示信息的第四个字开始到行末的字幕清除，保留前三个中文字"请输入"，其中语句800把光标横线定位在第4个汉字的起始位置，语句810清除字幕。在输入栏标题时，语句300,320，380中的中文提示接在"请输入"三个字之后，显示组成完整的提示信息。语句310，360，370，390分别把输入的数据登记到屏幕上表格的相应表目下面。

4. CALL CL

这个子程序清除光标横线所在行的全部字幕。

例4　上例中如果要求在屏幕上第一行显示输入操作的提示信息，当输入出错时，在第一行显示错误性质并提示用户应如何处理。可以在程序中使用CALL CL语句，例3程序修改如下：

```
275 FOR K=1 TO 5:POKE 214, 0:PRINT K:NEXT K
```

```
280 POKE 215, 0:POKE 214, 2
290 INPUT "栏长度"; LN
300 POKE 214, 5:POKE 215, 4:PRINT LN:GOSUB 800
310 INPUT "类型 0:文字 1:数字" LM
320 ON LM+1 GOTO 340, 350
330 Y$="类型选择错!": GOSUB 900:GOTO 310
340 POKE 214, 11:POKE 215, 4:PRINT "文字": GOSUB 800:GOTO 360
350 POKE 214, 11:POKE 215, 4:PRINT "数字": GOSUB 800
360 INPUT "栏标题:" ;C$
370 POKE 214, 17:POKE 215, 4:PRINT C$:GOSUB 800
380 END
      ⋮
      ⋮
800 POKE 215, 0:POKE 214, 2
810 CALL 49959
820 RETURN
      ⋮
      ⋮
900 POKE 214, 2:POKE 215, 0:CALL 49959
910 PRINT Y$;"请按 CTRL-X 后重新输入"
920 GET G$
930 IF G$=CHR$(24) THEN GOSUB 800:RETURN
940 GOTO 920
```

程序运行到语句 310 时，屏幕上第一行显示提示信息："类型 0:文字 1:数字"，它提示用户，若这一栏的内容是文字(如姓名一栏)，按数字键 0 回答，若是数字(如年龄一栏)，按数字键 1 回答。若用户回答的不是"0"或"1"，则转去执行语句 900 开始的子程序，先把屏幕第一行清除，接着执行语句 910，在第一行显示"类型选择错！请按 CTRL-X 后重新输入"，提示用户处理方式。当用户按了CTR-X后转去执行语句 800 开始的子程序，再次清除第一行提示错误性质及指示处理方式的信息，然后返回语句 310，重新在第一行显示输入类型的提示信息，等待用户再次输入正确的选择。

5. CALL CF

这个子程序的功能和英文指令 ESC-F 的功能相同。用以清除光标横线后面直到屏幕右下角的全部字幕。

如果要在屏幕上设计一个字幕，屏幕上部是一个固定字幕，屏幕下部从某行开始的字幕是可变的。那末，每次要改变下部屏幕显示的内容时，可以使用POKE语句，先把光标横线移到这一行的最左面，然后使用 CALL CF 语句把这一行以下的全部字幕清除。

例 5　在例 4 中，若要输入10栏标题的长度、类型及标题名并且显示在表格的相应表目下，受屏幕显示区的限制，可以先显示前 5 栏的信息，在输入后 5 栏之前，把前 5 栏的显示信息全部清除，再在表目下显示后 5 栏的信息。程序中使用 CALL CF 语句。

```
200 CALL 49941
```

```
210 POKE 214, 3:POKE 215, 3
220 PRINT "栏长度"
230 POKE 214, 11:POKE 215, 3
240 PRINT "类型"
250 POKE 214, 17:POKE 215, 3
260 PRINT "栏标题"
270 CALL 49944
272 L=1
274I=L:J=4
276 FOR K=L TO L+4:POKE 214, 0:PRINT K:NEXT K
280 POKE 215, 0:POKE 214, 2
290 INPUT "栏长度:";LN(I)
300 POKE 214, 5:POKE 215, J:PRINT LN(I):GOSUB 800
310 INPUT "类型 0:文字 1:数字" LM(I)
320 ON LM(I)+1 GOTO 340, 350
330 Y$="类型选择错!": GOSUB 900:GOTO 310
340 POKE 214, 11:POKE 215, J:PRINT "文字":GOSUB 800:GOTO 360
350 POKE 214, 11:POKE 215, J:PRINT "数字":GOSUB 800
360 INPUT "栏标题:"; C$(I)
370 POKE 214, 17:POKE 215, J:PRINT C$(I):GOSUB 800
380 IF I=5 THEN POKE 215, 4:POKE 214, 0:CALL 49962:L=6:GOTO 274
390 IF I=10 THEN END
400 I=I+1:J=J+1:GOTO 280
        ⋮
        ⋮
800 POKE 215, 0:POKE 214, 2
810 CALL 49959
820 RETURN
900 POKE 214, 2:POKE 215, 0:CALL 49959
910 PRINT Y$;"按 CTRL-X 后重新输入"
920 GET G$
930 IF G$=CHR$(24) THEN GOSUB 800:RETURN
940 GOTO 920
```

语句 276 在第一列显示栏的编号，显示前 5 栏标题后，语句 380 中的 CALL CF 语句把这 5 栏标题全部清除，继续显示后 5 栏标题。

三、控制打印输出的 POKE 语句

在 C-PLUS Ⅱ-A 汉卡系统中如下的 POKE 语句控制打印输出，各个 POKE 位置与打印机、汉卡所在的扩充插座号码有关，下面以 CN 代表汉卡所在的扩充插座号码，PT 代表打印机所在的扩充插座的号码，并设汉卡插在 3号扩充插座上，打印机插在 1号扩充插座上，

即 CN=3，PT=1。

1．开关打印机的 POKE 语句

(1) POKE 1400+CN，PT

例如：POKE 1403，1

表示接通打印机，准备打印，相当于英文语言中的PR#1功能。

(2) POKE 1400+CN，0

例如：POKE 1403，0

表示关闭打印机，相当于英文语言中的 PR#0 的功能。

2．控制打印格式和字形的 POKE 语句

(1) POKE 1656+CN，0

例如：POKE 1659，0

表示打印机以大字横印。

例 1　以大字横印打印一个报表表头"同济大学教师名册"

```
10  POKE 1403，1
20  POKE 1659，0
30  PRINT "同济大学教师名册"
40  POKE 1403，0
```

打印结果如下所示。

同济大学教师名册

它定位在打印纸的最右面，系统规定，若对打印机字形和格式不加选择，机器自动选定为大字横印。当打印机没有被设定过其他的打印格式及字形，字形语句20不用，结果是一样的。反之，这个语句不能省略。

例 2　以大字横印打印一个报表表头，要求表头定位在打印纸的正中间，且在表头下面打印一道点划线。假设使用 MX-80 打印机，每行最多能打印 30 个大字横印汉字。程序如下：

```
]LIST
1000 POKE 1403，1
1010 POKE 1659，0
1020 GOSUB 1100
1030 PRINT "同济大学教师名册"
1040 GOSUB 1100
1050 FOR J=1 TO 16
1060 PRINT "-";
1070 NEXT J
1080 PRINT ""
1090 POKE 1403，0:END
1100 FOR J=1 TO 22
1110 PRINT "␣";
1120 NEXT J
1130 RETURN
```

程序中语句 1100 开始的子程序打印空格，使表头能定位在打印纸的正中间。

(2) POKE 1656+CN，1

例如：POKE 1659，1

表示打印机以大字直印。

例 3　在例 1 中，如果把 20 句改为 POKE 1659，1，则打印结果如下：

同济大学教师名册

(3) POKE 1656+CN，2

例如：POKE 1659，2

表示打印机以小字横印。

例 1 中的 20 语句改为 POKE 1659，2，则打印结果如下：

同济大学教师名册

例 4 用小学横印打印例 2 的程序清单，步骤如下：

]PR#3　(按 RETURN 键)；

]POKE 1403，1(按 RETURN 键，接通打印机。)；

]POKE 1659，2(按 RETURN 键，这一行在打印机的纸上打印出来。)

]LIST 1000，1130(按 RETURN 键，开始打印例 2 的程序清单，直到打印结束。)；

]POKF 1403，0(按 RETURN 键，关闭打印机，这一行仍然打印出来。)。

例 5　输入一个报表的各栏标题，表格的宽度小于 60 个汉字，使用 MX-80 打印机，要求以小字横印打印各栏标题。

```
10  INPUT "总栏数:";N:DIM C$(N)
20  FOR I=1 TO N: INPUT C$(I):NEXT I
30  POKE 1403, 1:POKE 2043, 120
40  POKE 1659, 2
50  FOR I=1 TO N
60  PRINT C$(I);"␣";
70  NEXT I
80  PRINT ""
90  POKE 1403, 0
100 END
```

语句 60 打印各栏标题，并在每两栏标题之间留一个空格。语句 POKE 2043，120 定义一行中最多可以打印的字数(在本节稍后叙述。)。

(4) POKE 1656+CN，3

例如：POKE 1659，3

表示打印机以小字直印。

例 1 中的语句 10 改为 POKE 1659，3，打印结果如下：

同济大学教师名册

例 6　打印一个报表，要求整个表格定位在打印纸的正中，表头以大字横印定位在表格正中，副表头以大字横印定位在表格左上角，栏标题及各报表行以小字横印，定位在表格正中。

由于在 C-PLUS Ⅱ-A 汉卡中不能使用 TAB 函数，所以报表打印输出时的定位工作必须用程序完成。为使各栏输出的内容对齐，应以一栏中可能出现的最长记录来取定各栏的设定长度。在下面编制的程序中

报表表头在标号为 3200的DATA 语句中给出并读入 A$；
报表副表头在标号为 3210 的 DATA 语中给出并读入B$；
报表的总栏数，总行数在语句 3220 中给出并读入 N，M；
各栏的设定长度在语句 3230 中给出并读入 LN(I)；
各栏的标题在语句 3240 及 3250 中给出并读入 C$(I)；
各个报表行在语句 3260～3290 中给出并读入 D$(I，J)。
本程序中报表栏共 10 栏，并给出两行报表内容。

```
1500 TL=0
1505 READ A$:HH$=A$:GOSUB 3100:LA=LH/2
1510 READ B$:HH$=B$:GOSUB 3100:LB=LH/2
1515 READ N, M:DIM LN(N):DIM C$(N):DIM D$(M, N)
1520 FOR I=1 TO N: READ LN(I):TL=TL+2*LN(I)+1:NEXT I
1525 FOR I=1  TO N
1530 READ C$(I):HH$=C$(I):GOSUB 3100
1540 IF LH=2*LN(I) THEN 1580
1550 FOR J=1 TO 2*LN(I)-LH:C$(I)=C$(I)+"␣":NEXT J
1580 NEXT I
1800 FOR I=1 TO M:FOR J=1 TO  N:READ D$(I,J):HH$=D$(I, J):GOSUB
     3100
1810 IF LH=2*LN(J) THEN 1830
1820 FOR K=1  TO 2*LN(J)-LH:D$(I, J)=D$(I, J)+"␣":NEXT K
1830 NEXT J
1850 NEXT I
2000 POKE 1403, 1:POKE 1659, 0:POKE 2043, 120
2010 L1=(60-2*LA)/2:GOSUB 3000
2020 PRINT A$
2030 GOSUB 3000:L2=2*LA:GOSUB 3050
2040 L1=(120-TL)/4:GOSUB 3000
2050 PRINT B$
2060 GOSUB 3000:L2=2*LB:GOSUB 3050
2070 POKE 1659, 2
2075 L1=(120-TL)/2
2080 GOSUB 3000
2090 FOR I=1 TO N
2100 PRINT C$(I);"␣";
2110 NEXT I
2120 PRINT ""
2130 GOSUB 3000
2140 L2=TL:GOSUB  3050
2150 FOR I=1 TO M: GOSUB 3000
```

```
2155 FOR  J=1 TO N
2160 PRINT D$(I, J);"␣";
2170 NEXT J
2180 PRINT ""
2185 NEXT I
2190 POKE 1403, 0:POKE 1659, 0:POKE 1915, 0:POKE 1787, 4:END
3000 FOR J=1 TO L 1
3010 PRINT"␣";
3020 NEXT J
3030 RETURN
3050 FOR J=1 TO L2
3060 PRINT"-";
3070 NEXT J
3080 PRINT ""
3090 RETURN
3100 LH=0:JJ=0
3110 JJ=LEN(HH$)
3120 IF JJ=0 THEN LH=0:RETURN
3130 FOR II=1 TO JJ
3140 BB$=MID$(HH$, II, 1)
3150 KK=ASC(BB$)
3160 IF KK<97 AND KK<>32 OR KK<> 121 THEN LH=LH+1: GOTO 3180
3170 IF KK=32 THEN LH=LH+2
3180 NEXT II
3190 RETURN
3200 DATA "同济大学教师名册"
3210 DATA "建筑系"
3220 DATA 10, 2
3230 DATA 4, 2, 4, 4, 4, 4, 4, 12, 10, 2
3240 DATA "姓名", "性别", "出生年月", "政治面貌", "文化程度"
3250 DATA "职务", "职称", "家庭地址", "何年何月受何奖惩", "婚否"
3260 DATA "王林", "男", "35,17,12", "党", "大学"
3270 DATA "室主任", "副教授", "延安路 100 号", "", "已"
3280 DATA "陈红", "女", "42.5.6", "", "大学"
3290 "", "讲师", "北京路 50 号","","已"
```

程序中语句 3100～3190 是计算一个包含汉字的字符串长度的子程序，它给出了这个含有汉字的字符串输出时占据了几个英文字母的打印位置；

语句 3000～3030 是打印空格子程序；

语句 3050～3090 是打印点划线子程序；

语句 1500～1850 读入A$,B$，C$(I)，D$(I,J)，计算各个数据的长度，当 C$(I)，D$

(I，J)中每一个数据与其所在栏设定的长度不等时，必须以空格补齐，使每一栏的各个数据等长，打印时各栏才能对齐。并由各栏设定长度计算报表一行的总长度 TL(它还计入每两栏之间所留的一个空格数目)；

语句 2000 接通打印机，定义大字横印及每行最多可打印 60 个汉字；

语句 2010 根据表头的长度 LA，计算打印纸两边应留的空格数，使表头能定位在打印纸正中；

语句 2030 在表头下面打印点划线；

语句 2040 由一行的总长度 TL，计算在表格两边应留出的空格数，使整个报表定位在打印纸正中；

语句 2050 使副表头打印在报表左上角；

语句 2060 在副表头下打印点划线；

在打印栏标题及报表行之前由语句 2070 设定小字横印。

语句 2080～2120 打印栏标题；

语句 2140 在栏标题下面打印点划线；

语句 2150～2185 打印各报表行；

语句 2190 关闭打印机，结束打印程序。

语句 2000 中的 POKE2043，120是定义一行打印字数的POKE语句，请看后面的叙述。)

这是一个通用的报表打印程序，打印不同的报表，只要修改 DATA 语句中的报表数据即可。若要打印包含空格符的字符串须修改语句3100～3190的子程序；读者不妨一试。

3．定义打印时行、列间距的 POKE 语句

(1) 定义打印时列间距的 POKE 语句

格式：POKE 1912＋CN，N

例如：POKE 1915，N

定义打印时每两个英文(包括数字、符号)字之间的距离为 N 个点的间距。中文字之间的距离为 2N 个点的距离。

例如：N＝4，则英文与英文字母之间的距离为 4 点，中文与中文字之间的距离为 8 点，由于中文字点阵为 16×16，所以这里两个中文之间的距离为一个英文字的大小，也即半个汉字的大小，两个英文字母间的距离为半个英文字母的大小。一个英文字与中文字间的距离也为半个英文字的大小。

如果不设定N值，则N自动设定为 0。

例 2 的程序，不设定N值及设定 N＝4 时打印的程序清单分别如下：

不设定 N

```
]LIST
1000 POKE 1403，1
1010 POKE 1659，0
1020 GOSUB 1100
1030 PRINT ″同济大学教师名册″
1040 GOSUB 1100
1050 FOR J＝1 TO 16
1060 PRINT ″－″；
```

```
1070 NEXT J
1080 PRINT ""
1090 POKE 1403, 0:E
1100 FOR J=1 TO 22
1110 PRINT "␣";
1120 NEXT J
1130 RETURN
```

N=4

```
]POKE1915, 4
]POKE2043, 120
]LIST
1000 POKE1403, 1
1010 POKE1659, 0
1020 GOSUB1100
1030 PRINT"同 济 大 学 教 师 名 册"
1040 GOSUB1100
1050 FORJ=1TO16
1060 PRINT "-";
1070 NEXTJ
1080 PRINT""
1090 POKE1403, 0:E
1100 FORJ=1TO22
1110 PRINT "␣";
1120 NEXTJ
1130 RETURN
```

（2）定义打印时行间距的语句

格式：POKE 1784+CN，N

例如：POKE 1787，N

定义打印时行与行之间的距离为N个点的间距，如果不定义N值，则自动设定为N=4。

例7　在例6中，要求根据报表一行的总长度确定大字横印或小字横印报表栏标题及报表行。若报表宽度大于30个汉字时小字横印，小于30个汉字时大字横印，大字横印时行与行间距离为4个点的间距，小字横印时行与行间的距离为两个点的间距。

只要在例6中加入语句2035，2065，并修改2070语句即可。

```
2035 IF TL<60 THEN L1=(60-TL)/2:GOSUB 3000:GOTO 2050
2065 IF TL<60 THEN 2080
2070 POKE 1659, 2:POKE 1787, 2
```

4．定义一行打印字数的POKE语句

格式：POKE 2040+CN，N

例如：POKE 2043，N

N定义打印机每行允许打印的字数，它以英文字为单位，中文字占两个英文字位置。例

如N=80，表示每行可打印80个英文字，或40个中文字。MX-80打印机一行最多可以打印60个中文字，MX-100打印机可打印120个中文字，如果不定义N值，则自动设定N=50。

这里要注意两点：

（1）N指的是在列间隔定义为0时打印机每行允许打印的字数；

（2）每行允许打印的字数包括列间距的点值在内，所以当POKG 1912+CN，N中的N≠0时，一行中所能打印的实际字符数（包括由这些字符，汉字组成的项与项之间的空格在内）不再是POKE 2043，N中定义的N值，而应该扣除列间距所占的位置。

当设定N=25时，例2程序清单打印结果如下：

```
]POKE 2043, 25
]LIST
1000 POKE 1403, 1
1010 POKE 1659, 0
1020 GOSUB 1100
1030 PRINT "同济大学教师
名册"
1040 GOSUB 1100
1050 FOR J=1 TO 16
1060 PRINT "-";
1070 NEXT J
1080 PRINT ""
1090 POKE 1403, 0:E
1100 FOR J=1 TO 22
1110 PRINT"␣";
1120 NEXT J
1130 RETURN
```

5．控制自动跳页功能的POKE语句

格式：POKE 1528+CN，12

例如：POKE 1531，12

相当于在英文APPLE SOFT BASIC中PRINT CHR$(12)的功能，即自动跳页功能。在汉字系统中，首先必须将12加载到1528+CN的位置，使用跳页功能时，再用CALL BS+45(BS=49152+256*CN)语句调用跳页功能子程序。

例8　在例7中，要求打印的份数可以由用户自己选择，每打印一份报表，自动跳过一页。只需在例7程序中增加语句1400，1410,2200,2210并修改语句2190。完整程序如下：

```
1400 INPUT "打印几份";V
1410 U=1:POKE 1531, 12
1500 TL=0
1505 READ A$:HH$=A$:GOSUB 3100:LA=LH/2
1510 READ B$:HH$=B$:GOSUB 3100:LB=LH/2
1515 READ N, M: DIM LN(N):DIM C$(N):DIM D$(M, N)
1520 FOR I=1 TO N:READ LN(I):TL=TL+2*LN(I)+1:NEXT I
```

```
1525 FOR I=1  TO N
1530 READ C$(I):HH$=C$(I):GOSUB 3100
1540 IF LH=2*LN(I) THEN 1580
1550 FOR J=1 TO 2*LN(I)-LH:C$(I)=C$(I)+"␣":NEXT J
1580 NEXT I
1800 FOR I=1 TO M:FOR J=1 TO N:READ D$(I, J): HH$=D$(I, J):GOUB
     3100
1810 IF  LH=2*LN(J),THEN 1830
1820 FOR K=1 TO 2*LN(J)-LH:D$(I, J)=D$(I, J)+"␣":NEXT K
1830 NEXT J
1850 NEXT I
2000 POKE 1403, 1:POKE 1659, 0:POKE 2043, 120
2010 L1=(60-2*LA)/2:GOSUB 3000
2020 PRINT A$
2030 GOSUB 3000:L2=2*LA:GOSUB 3050
2035 IF TL<60 THEN L1=(60-TL)/2:GOSUB 3000:GOTO 2050
2040 L1=(120-TL)/4:GOSUB 3000
2050 PRINT B$
2060 GOSUB 3000:L2=2*LB:GOSUB 3050
2065 IF TL<60 THEN 2080
2070 POKE 1659, 2:POKE 1787, 2
2075 L1=(120-TL)/2
2080 GOSUB 3000
2090 FOR I=1 TO N
2100 PRINT C$(I);"␣";
2110 NEXT I
2120 PRINT ""
2130 GOSUB 3000
2140 L2=TL:GOSUB 3050
2150 FOR I=1 TO M:GOSUB 3000
2155 FOR J=1 TO N
2160 PRINT D$(I, J);"␣";
2170 NEXT J
2180 PRINT ""
2185 NEXT I
2190 IF U=V THEN POKE 1403, 0:END
2200 U=U+1:CALL 49152+256*3+45
2210 GOTO 2000
3000 FOR J=1 TO L1
3010 PRINT "␣";
```

```
3020 NEXT J
3030 RETURN
3050 FOR J=1 TO L2
3060 PRINT "-";
3070 NEXT J
3080 PRINT ""
3090 RETURN
3100 LH=0:JJ=0
3110 JJ=LEN(HH$)
3120 IF JJ=0 THEN LH=0:RETURN
3130 FOR II=1 TO JJ
3140 BB$=MID$(HH$, II, 1)
3150 KK=ASC(BB$)
3160 IF KK<97 AND KK<>32 OR KK>121 THEN LH=LH+1:GOTO 3180
3170 IF KK=32 THEN LH=LH+2:GOTO 3180
3180 NEXT II
3190 RETURN
```

DATA 语句同本节例6，不再列出。

四、控制显示中文字母的 POKE 语句

1. POKE 253，0

这个功能与 CTRL-D 相同，在显示中文字时，左下角不会产生小的中文字母。

2. POKE 253，255

恢复原始状态，即左下角显示中文字母小字。

这两个语句可以在程序中使用，处于 POKE 253，255 和 POKE 253，0 两个语句之间的一段程序执行时，凡是遇到中文输出，组成每个汉字的中文字母都会在左下角显示出来。

第九节　系统函数及自定义函数

一、系统函数

1. PEEK、POKE

APPLE Ⅱ 的每个存贮单元可以贮存一个 0～255 之间的数值。所有程序或数据都被转换成一系列的数字存贮在存贮器中，汉字信息是以组成汉字的中文字母 ASC Ⅱ 码方式存贮的。存贮单元地址可以使用十进制表示，也可以使用十六进制表示，若用十六进制表示，须在其前面加一个 $ 符号。通常，小于 32767 的地址用正值表示，大于 32767 的地址用负值表示，它由地址的正值减去 65536 得到。

系统提供 PEEK 功能从内存某个单元中读出信息，还提供了 POKE 功能把信息存到内存的任一单元中去。

(1) POKE

格式： POKE 内存地址，数值

内存地址是随机存贮器RAM地址，以十进制表示；数值范围是 0～255。

其功能是把一个 0～255 之间的数值存放到指定的内存 RAM 的一个单元中。要把一个汉字存放到内存某一区域，应先求得组成汉字的各个中文字母的 ASCⅡ码，再用 POKE语句把各个码值顺序存入内存。

例 1　试在内存 898 单元中写入 8 个 bit 的 1。

```
10 POKE 898, 255
```

例 2　试把“日”字存入内存 899 单元。

```
10 POKE 899, ASC("日")
```

例 3　试把“部”字存入内存 910 单元开始的存贮区中。

```
10 A = LEN("部")
20 FOR I = 1 TO A:POKE 910 + I, ASC(MID$(A$, I, 1)):NEXT I
```

(2) PEEK

格式：PEEK(内存地址)

内存地址以十进制表示。

一个 PEEK 语句仅能读出一个存贮单元中的数值，其值范围为 0～255。如果要读出存放在某一存贮区中的汉字，必须先用 PEEK 语句把存贮区中每一个中文字母的 ASCⅡ码读出来，再转换成相应的中文字母并组合成一个汉字。

例 4　试读出内存 899 单元的内容并赋给变量 A。

```
5 POKE 899, ASC("日")
10 A = PEEK(899)
20 PRINT A
RUN
97
```

例 5　试把组成“部”字的四个中文字母“卜”、“口”、“弓”、“中”的 ASCⅡ码顺序存放到 24576 单元开始的内存区域中，并重新从这个区域中读出来组成“部”字。已知这四个中文字母的 ASCⅡ码分别为 121，114，110，100。程序如下：

```
10 FOR I = 1 TO 4:INPUT L:POKE 24575 + I, L:NEXT I
20 H$ = ""
30 FOR I = 1 TO 4: H$ = H$ + CHR$(PEEK(24575 + I)):NEXT I
40 H$ = H$ + ""
50 PRINT H$
60 END
RUN
?121
?114
?110
?108
部
```

2. FRE

FRE 函数指出字符串存贮区目前有多少剩余空间。它可用在 PRINT 语句中。如：

10 PRINT FRE(1)

括号中的参数值是任意的。

程序执行时，如果一个字符串改变了，而原有值仍然保留。这样，内存中有许多无法使用的零散空间。使用 A = FRE(0)语句，系统会把所有无法使用的零散空间结合成一体，释放出来以供使用，一般用于各程序段起始位置。

3．USR

格式：USR(A)

A 为表达式。

USR 函数能将 A 传送到机器语言编写的子程序中去。经计算后，结果置于浮点累加器中，再把累加器的值传回 APPLE SOFT，USR 函数的值就是浮点累加器的值。

USR 函数执行过程中，要求在主机的 $OA～$OG 单元中放一条机器指令 JMP（代码为 4C)，以跳到机器语言子程序的起始位置去。机器语句子程序的最后一条指令为 RTS(代码为 60)，用以返回 APPLE SOFT。这些工作须在监控系统下完成。例如：

```
]CALL-151 RETURN
* 0A:4C 0003 RETURN
* 0300:60 RETURN
* CTRL-C RETURN
]PRINT USR(8)*3 RETURN
24
```

用 CALL-151 命令进入监控系统后，在 $OA～$OG 单元中存进 JMP$300，在 $300单元存进RTS指令（操作方法见第五章第三节），当执行 PRINT USR(8)*3 时，数字 8 被存进累加器中，监控程序使机器转到 $OA 取出 JMP$300 指令，并跳到 $300 去执行返回指令 RTS，返回 APPLE SOFT，这时 USR 函数的值就是累加器中的值 8，APPLE SOFT 把 8 乘以 3 得 24 并显示。

二、自定义函数语句(DEF FN)

APPLE SOFT 除了提供前面已叙述过的基本算术函数以外，还允许用户使用自定义函数语句定义一些算术函数，但不允许定义字符串函数。

例如：计算圆面积公式为 πR^2，如果需要多次使用，可以把它定义为一个函数，以供随时调用。

自定义一个函数使用语句

```
10  DEF FN  P(X) = 3.14159*X*X
```

函数名为 P，其中 X 为自变量，它是一个虚设的变量。可以是任一简单变量，而且只允许有一个变量，它和主程序中同名的变量彼此独立。

调用自定义函数时，须在该函数名前加 FN。

例如：

```
30  PRINT FN  P(A)
```

这时虚设变量换成实际变量、常数或表达式。在调用之前，变量必须先赋值。调用实例如下：

```
10  DEF FN P(X) = 3.14159*X*X
```

```
20 INPUT A
30 B=FN P(A)+5
40 PRINT B
```

如果输入的 A=2，执行这个程序后输出的函数值为 17.56636。

第十节 光标控制及屏幕显示格式语句

在英文 BASIC 语句中提供了确定光标当前水平位置的 POS 函数、控制光标移动的 HTAB，VTAB 语句以及确定屏幕显示格式的 INVERSE，NORMAL，FLASH 语句，清除屏幕显示的 HOME 语句等。这些语句在配置 C-PLUS Ⅱ-A 汉卡时，不能使用。但使用超级汉卡时，除了 FLASH 语句外，其余语句仍保留原有功能。详见第六章叙述。

一、确定光标位置函数(POS)

POS 函数确定光标当前的水平座标值。若光标在屏幕最左方，水平座标值为 0。

格式：POS(0)

括号中的 0 没有什么意义，但必须写上。

POS(0)的值为 0～39。

例如：]?"CURSOR␣POSITION␣IS␣";POS(0)

PRINT 命令执行后，在屏幕上显示

　　CURSOR␣POSITION␣IS␣19

这里 POS(0)值为 19，它指出了在显示 CURSOR POSITION IS 之后，光标位于第 19 列，这是因为包括空格在内的显示信息共 19 个字符，而且 POS 的位置是从 0 算起。

值得注意的是 TAB 的位置从 1 算起，SPC 位置是从 0 算起，所以

]PRINT TAB(23);POR(0)　　显示 22

]PRINT SPC(23);POS(0)　　显示 23

二、光标控制语句(HTAB，VTAB)

在 PRINT 语句中，使用逗号及分号可控制光标的位置，在 APPLE SOFT 中 SPC 及 TAB功能也对光标定位非常有用。如果要把光标移到屏幕的任何位置，则要使用垂直移动光标的 VTAB 语句及水平移动光标的 HTAB 语句。

1. VTAB

格式：VTAB A

A 是表达式，A 的值表示垂直座标值，为 1～24。其中屏幕最上方为第一行，最下方为第 24 行。如 A 的值为 N，则 VTAB 使光标垂直移动到第N行。

如果 A 的值超出 1～24，机器报错：

? ILLEGAL QUANTITY ERROR

2. HTAB

格式：HTAB A

A 为表达式，表示水平座标值，其值为 1～255，屏幕最左边定义为 1。如 A 的值为 N，HTAB 使光标水平移动到第N列。当 A 的值为 1～40，光标出现在本行，A 的值为41～80，

光标移到下一行。以此类推，

其中 HTAB 0 使光标移到本行的第 256 坐标值位置上。

如果 A 的值为负或大于 255，机器报错：

?ILLEGAL QUANTITY ERROR

例 1　使屏幕上第 10 行，第 10 列开始显示如下信息：

1　NAME

2　STREET

3　CITY

程序如下：

```
10 HOME
20 VTAB 10:FOR I=1 TO 3:HTAB 10:PRINT I:NEXT I
30 VTAB 10:HTAB 12:PRINT "NAME"
40 HTAB 12:PRINT "STREET"
50 HTAB 12:PRINT "CITY"
60 END
```

三、屏幕显示格式语句

APPLE Ⅱ 可以在屏幕上以三种显示格式显示信息，即正常显示方式，反相显示方式及闪烁显示方式。

1. HOME

格式：HOME

HOME 语句清除整个屏幕显示，并把光标移到屏幕的左上角。

2. NORMAL

格式：NORMAL

它使屏幕显示回到正常显示方式，即以白字黑底显示信息。

3. INVERSE

格式：INVERSE

它使屏幕显示变成反相显示方式，即以白底黑字显示信息。可以使用 NORMAL 使机器回到正常显示方式。

例如：输入命令

```
]INVERSE
]?"BLACK ON WHITE"
```

反相显示

BLACK ON WHITE

```
]NORMAL
]?"WHITE ON BLACK"
```

正常显示

WHITE ON BLACK

4. FLASH

格式：FLASH

执行 FLASH 后，屏幕上输出的信息以正常和反相两种显示方式交替显示，产生闪烁。输入NORMAL 可以使机器回到正常显示方式。

例 2　在例 1 中，要求以反相显示方式显示数字 1，2，3，以闪烁方式显示 NAME，以正常方式显示 STREET，CITY。

程序如下：

```
10  HOME
20  INVERSE:VTAB  10:FOR I=1 TO 3:HTAB  10:PRINT I:NEXT I
30  FLASH:VTAB  10:HTAB  12:PRINT "NAME"
40  NORMAL:HTAB  12:PRINT "STREET"
50  HTAB  12:PRINT "CITY"
60  END
```

四、设定输出速度语句(SPEED)

格式：SPEED = A

A 为表达式，其值为 0～255。

SPEED 语句设定信息在屏幕上的显示速度或输出到其它外设的传送速度。A = 0，显示的速度最慢，A = 255，显示的速度最快。开机时初始设定 A = 255，如果 A 的值超出范围，机器报错：

? ILLEGAL QUANTITY ERROR

例如：

```
10  INPUT "速度码=；V
20  SPEED=V
30┐
 ⋮ │
 ⋮ ├PRINT 语句
 ⋮ │
80┘
90  SPEED=255
```

标号为 30～80 的 PRINT 语句中的信息，以和所选择的速度码 V 相应的速度显示。语句 90 恢复正常的显示速度。

这个语句在汉字系统中功能不变。

第十一节　其 它 语 句

一、设置输入输出通道的语句

APPLE SOFT 使用输入、输出设备是通过语句 IN#N，PR#N 实现的。其中 N 指外设所插的外设扩充插座的编号。N 取值范围为 0～7。例如，打印机控制卡插在 1号插座上，N = 1；软盘驱动器控制卡插在 6号扩充插座上，N = 6，监视器屏幕作为输出设备，键盘作为输入设备，系统把它们的 N 值定为 0，机器开机或复位(RESET)时，系统设定 N = 0。

1. PR#N

使用 PR#N，可以选择一个外部设备作为输出设备。例如，在英文系统中，使用打印机时，必须输入 PR#1，使输出通道和打印机接通。

若在程序之后输入

```
]PR#1
]LIST
```

将打印这个程序清单。打印结束后，输入]PR#0，使输出通道和打印机断开并重新和监视器接通。

同样可以在程序中使用 PR#0、PR#1。例如：

```
10  PR#1
20  PRINT "APPLE"
30 PR#0
RUN
APPLE
```

如果 DOS 已加载，PR#N 也可以在程序中作为 DOS 命令使用。例如：

```
10  D$=CHR$(4):PRINT D$;"PR#1"
20  PRINT "APPLE"
30  PRINT D$;"PR#0"
```

执行这个程序，在打印纸上打出"APPLE"。

当汉卡插在3号插座，使用 PR#3，选择汉卡作为输出设备，进入汉字系统。

2. IN#N

使用 IN#N，可以选择一个外部设备作为输入设备，例如在 APPLE SOFT 工作方式下要启动软盘驱动器可以输入 IN#6。

如果指定的插座上没有外设控制卡，那末 APPLE Ⅱ 将被锁住，必须复位后再重新选择。

二、CALL 语句

CALL 语句调用机器语言子程序。

格式：CALL 入口地址

入口地址须是十进制地址。

入口地址是机器语言子程序在内存中的起始地址。CALL 语句不仅可调用 RAM 中的子程序，也可调用 ROM 中的子程序。用户可以方便地调用监控系统中现成的子程序，也可以自己编写机器语言子程序，存于存贮器中，然后在 BASIC 程序中用 CALL 语句调用。

三、ONERR GOTO 及 RESUME 语句

格式：ONERR GOTO 标号

在数据输入或读一个盘文件时，由于输入的错误或读盘文件程序设计不正确会引起程序中断。为了避免因小错误而中止程序执行，APPLE SOFT 提供了 ONERR GOTO 语句。

此语句必须放在程序前面，程序执行时，如发现有错，将跳到 ONERR GOTO 所指明的标号去执行，同时一个描述错误性质的数码被存到224单元，可以用 PEEK(224)语句取出这个码，分析错误性质。

通常在这语句指出的标号处设计一段处理错误的程序，在程序结束前，写上 RESUME

或 GOTO 语句，以便返回到发生错误的语句。并重新执行这个语句。

若要返回正常的 APPLE SOFT 查错状态，可以使用语句

POKE 216，0

例如

```
10  ONERR GOTO 50
20  INPUT A
30  PRINT A
40  END
50  PRINT "请输入数字"
60  GOTO 20
```

执行语句 10 的 ONERR GOTO 后，如果输入一个字符，程序跳到语句 50，指示必须输入数字，然后回到出错的语句20处重新执行输入操作。语句60可以用 RESUME 代替。

如果不写语句 10，系统处于 APPLE SOFT 正常的查错状态，输出"?REENTER"提示信息。

例如：

```
10  ONERR  GOTO 50
20  INPUT A
30  PRINT A
40  END
50  PRINT "请输入数字"
60  PRINT PEEK(224):POKE 216, 0
70  RESUME
```

上例程序中，当第一次输入一个字符，输出提示信息"请输入数字"，然后读出出错码，并返回 APPLE SOFT 查错状态。当第二次输入出错，系统提示 REENTER。若输入正确，即输入了一个数字，则正常结束。

对于开平方根这样的函数，要求被开方数必须大于或等于 0，如果输入一个负数，则出现出错信息：

?ILLEGNL QUANTITY ERROR IN   N

程序中断。这时 ONERR GOTO 语句可以避免程序中断。

例如：

```
10  ONERR GOTO 50
20  INPUT X
30  Y=SQR(X)
40  PRINT Y:END
50  PRINT "输入正数或零"
60  INPUT X
70  RESUME
```

错误出现在语句30，所以执行语句60，重新输入X，然后回到语句30重新执行开平方过程。

四、CLEAR 语句

CLEAR 语句将程序中所有的数值为 0，所有的字符串清除为空串。

例如：

```
10  A$="年令"：A=30
20  PRINT A$;"␣";A
30  CLEAR
40  PRINT A$;"␣";A
50  END
```

程序执行到语句 20，显示"年龄␣30"，语句 30 把变量A 及 A$ 清除，执行语句 40，显示"0"。

第四章　DOS 操作系统

APPLE Ⅱ PLUS 微机上配置了一个磁盘操作系统，目前用得最多的是 DOS 3.3 版本，它的主要功能是完成系统的操作管理，特别是磁盘的读、写操作和磁盘文件的各项管理操作。

DOS 3.3 是用 6502 机器指令编写的，当系统引导后，它被装入内存，约占 10K 存贮空间，整个 DOS 系统常驻内存。

在 DOS 操作系统支持下的有 APPLE SOFT BASIC，整数 BASIC 语言也支持中文 APPLE SOFT BASIC 语言。

DOS 3.3 提供了近三十条操作命令，由于在汉字系统中不能使用整数 BASIC 语言，INT、CHAIN 等命令不能使用，FP 命令不起作用。

DOS 3.3 系统主盘中保存有 DOS 操作系统，在操作系统引导到内存之前，机器拒绝执行所有 DOS 命令，初始化后的软盘上装有 DOS 操作系统的副本，它们都可以用来引导 DOS。

DOS 3.3 版本和老的 DOS 3，DOS 3.1，DOS 3.2 等主要差别在于对扇区的划分，DOS 3.3 提供了对 16 个扇区磁盘的操作，而老版本使用划分 13 个扇区的磁盘。

第一节　磁盘信息的组织

一、软盘和软盘驱动器

APPLE Ⅱ 使用 $5\frac{1}{4}''$ 软盘驱动器，软盘驱动器控制卡一般接在 6 号扩充插座上，一个控制卡可驱动两个软盘驱动器，分别称为 1 号驱动器和 2 号驱动器，并以 D1 和 D2 表示，当初始起动时，自动设定为 D1，即 DOS 命令对 1 号驱动器进行操作。

APPLE Ⅱ 配置的软盘是单面单密度 $5\frac{1}{4}''$ 软盘，每个盘划分为 35 个道，DOS 提供对 16 扇区的磁盘的操作，所以每道划分为 16 个扇区。每个扇区可记录 256 字节的用户信息，整个磁盘的存贮容量为 143 K 字节。

在经过初始化的软盘上，第 0～2 道上存有 DOS 3.3 操作系统程序的副本，第 17 道上存放目录表等磁盘管理信息，其余的 31 道供用户存放程序文件及数据文件，所以，用户实际可使用的最大盘空间是 496 个扇区。

用户文件使用磁盘空间，系统按如下顺序进行分配，使用扇区时，从第 15 扇区向第 0 扇区依次分配，使用磁道时，先从第 18 道开始顺序使用到第 35 道，然后再从第 16 道顺序使用到第 3 道。

二、盘文件类型及其存贮格式

用户程序及数据都以文件方式存贮在盘上，DOS3.3 能直接建立的文件类型只有四种，

它们的名称及在目录表中所使用的代码见表 4-1-1。

表 4-1-1

代　　码	文　件　类　型
A	APPLE SOFT BASIC文件
I	整数 BASIC 文件
T	正文文件
B	二进制文件

1. BASIC 文件

此文件存放 BASIC 程序，按 APPLE SOFT BASIC 程序及整数 BASIC 程序分成两种类型的 BASIC 文件，在汉字系统中不能使用整数 BASIC 语言，所以只能是A型文件。

在盘中的存贮格式：

长度　程序的内存映象

长度指该程序占用内存的字节个数，存在头两个字节中，长度的低位放在前面，文件的其余空间存贮这个程序，以 ASCⅡ 字符形式存放。APPLE SOFT BASIC 的保留字都用数字代表，每个数字也使用相应的 ASCⅡ 码存放。

2. 正文文件

此文件一般是存放数据信息的文件，有两种不同型式的正文文件，随机文件和顺序文件。

正文文件都由记录组成，每个记录之间用 RETURN (CR)分隔，并以 $00 标志文件结束。

正文文件的存贮格式：

记录 1CR 记录 2CR……CR00

顺序文件各记录是不定长的，记录与记录之间在存贮时是相互紧接的，仅以一个 CR 来判别一个记录的结束位置，信息一般以顺序方式读写，随机文件各记录是定长的，因而可从任一指定的记录处开始读写。

正文文件中的信息都以可打印的 ASCⅡ 字符形式存贮的，在汉字系统中包括中文字母，每一个字符占一个字节。

包含有中文资料的正文文件读到主存后，能直接以中文显示这些资料。

3. 二进制文件

它是内存中二进制信息的副本，把内存中从某个地址开始的一段信息保存到盘上就生成二进制文件，通常这个信息是一般机器代码程序、二进制数据或内存第二页高分辨率图形区的映象。用户也可以把中文 BASIC 编写的中文信息处理程序作为二进制文件保存及调用。详见后面的叙述。

二进制文件存贮格式：

地址　长度　内存二进制信息映象

文件的头两个字节贮存这段信息原来在内存中的起始地址。低位地址放在前面，接下两个字节存贮内存映象的字节个数，文件的其它空间存放二进制信息。

第二节　DOS 操作系统的引导及盘片的初始化

一、DOS 操作系统的引导

由厂家提供的DOS 3.3操作系统盘，即主盘可以在有 16 K 以上 RAM 的任何 APPLE Ⅱ 机上引导 DOS，用 DOS 3.3 进行格式化后的盘（或称从盘）上有 DOS 副本，也可以引导 DOS。当 DOS 系统引导时，将 DOS 从盘上装入内存，DOS 在盘上共占 37 个扇区，0 道、1 道都占用 16 个扇区，第 2 道上占 5 个扇区，它装入内存的最高 RAM 存贮区域。在汉字系统中，它装入到主存 \$9600～\$BFFF 区域。

引导操作系统方法如下：

1．开机引导

把主盘或从盘放入 1 号驱动器，开机后进入监控系统的自启动程序入口，监控程序依次扫描各个扩充插座，如果插座上存在驱动器控制卡，则执行盘引导程序，进入盘片上的 HEL. LO 程序，屏幕上显示：

DOS VERSION 3.3　　　　　　　　08/25/80

APPLE Ⅱ　PLUS OR ROMCARD SYSTEM MASTER

2．执行引导命令

开机时，如果没有盘片放入驱动器，监控系统控制进入 BASIC，输入命令 PR#6 或 IN #6，也可以进入监控系统程序的自启动入口，引导过程和开机引导相同，在监控系统中执行 6 CTRL P 或输入 C 600 G 命令也可以引导 DOS。

DOS 引导时分三级进行，第一级引导程序是盘驱动器控制卡上固化的 ROM 程序，它读入盘上 0 道 0 区上的第二级引导程序并执行，第二级引导程序执行时读入磁盘驱动程序（RWTS），完成磁盘扇区的读写及格式化操作等，最后一级引导程序装入整个 DOS。

主盘上的 DOS 在引导时可以自动重定位，机器先将 DOS 放入相当于 16K 系统的最高 RAM 区域（\$1D00～\$4000），然后再移动到当前使用的系统最高 RAM 区域，所以主盘可以在具有 16K 以上 RAM 的任何 APPLE Ⅱ 机上引导 DOS。

利用从盘引导 DOS，机器直接把 DOS 装入与格式化的系统相同的最高 RAM 处，这就要求引导 DOS 的机器其 RAM 容量大于或等于该盘格式化的机器上的 RAM 容量，使用汉字系统时特别要注意这一点，在主盘中存放 MASTER，CREATE 程序，可以把从盘改成主盘形式。

DOS 引导结束，内存里存有 HELLO 程序，如果要输入新程序，应先用 NEW 命令清除 HELLO 程序。

二、盘片的初始化

空白的盘片必须经过初始化后才能使用，初始化工作必须在 DOS 引导进内存下工作，初始化的目的在于将 DOS 3.3 操作系统写入盘中第 0、1、2 道，完成划分磁道、扇区等规格化工作，初始化命令格式为 INIT f，其中 f 是以 BASIC 语言编写的程序的文件名，详见下面的 INIT 命令叙述，一般常用

INIT　　HELLO

这个程序的内容是初始化前内存中存放的程序，如果是在开机引导 DOS 之后直接初始化，即为 HELLO 程序，否则是初始化前内存遗留下来的任何程序，用户也可以根据需要写入适当的程序作为 HELLO 程序。

使用过的盘片也可以重新进行初始化，这时软盘上的所有资料都被清除，保存有用信息的盘片切不可作初始化。

一个盘如果要作为存放汉字信息处理程序及数据的介质，必须在具有不小于 48K RAM 的机器中进行格式化。格式化过程可以在英文 BASIC 下进行，也可以在中文 BASIC 下进行，两者输入相同的 INIT 命令，初始化时间较长，大约要半分钟左右，初始化结束，系统回答英文 BASIC提示符]或中文 BASIC 提示符]。

第三节　DOS 命令的格式及调用

APPLE Ⅱ 的 DOS 3.3 操作系统提供下列几种操作命令：

基本的程序存取命令：LOAD, SAVE;

运行命令为RUN。

二进制文件存取命令：BLOAD, BSAVE;

运行命令为BRUN。

正文文件操作命令：OPEN, WRITE, READ, CLOSE, APPEND, EXEC, POSITION;

盘文件处理命令：DELETE, LOCK, UNLOCK, RENAME;

监视命令：MON, NOMON;

辅助命令：CATALOG, FP, INT, INIT, PR#, IN#, VERIFY, MAXFILES, CHAIN。

在汉字系统中 INT, CHAIN 命令不能用，FP 命令无作用。

一、DOS 命令格式

大多数 DOS 命令名格式如下：

DOS 命令名 f[, Vv][, Ss][, Dd]

f：文件名，可以由 1～30 个字符组成，第一个字符必须是英文字母，在汉字系统中也可以是中文字母，其它字符是除逗点以外的任何字符，屏幕上不显示的控制字符也可以用在文件名中，用 CATALOG 命令查阅目录时，这些字符是看不见的，可用于对文件名加密码，但使用或清除文件时，这个控制字符仍需输入。

Vv：设定软盘片的卷号，V 值的范围是1～254，如果V取为 0 值或不设定V值，表示 DOS 采用和所使用的软盘片卷号相同的V值，设定V值，DOS 将检查磁盘驱动器中正在使用的盘片是否是操作者想要的那一片，如果V值和盘片上的卷号不同时，DOS 给出“VOLUME MISMATCH”的错误信息。V 初始设定为引导 DOS 时盘片的卷号。

Ss：选择驱动器控制卡所插的扩充插座号码，S 初始设定为引导 DOS 时磁盘控制卡所在的插座号码，通常是 6。使用过 S 值后，以后再次都定为这个值，直到重新设定S值为止，S 值范围 1～7。如果系统使用多个驱动器，可以将控制卡插到 1～7 中其它插座上，并用S值选择。

Dd：设定驱动器号码，d 初始值设定为 1，d 的范围为 1～2，每个控制卡带两个驱动器，一个为 D_1，另一个为 D_2，当驱动器多于 2 个时，必须用 Ss 和Dd 配合以选择驱动器。

例如： INIT HELLO，V254，S6，D1

或 INIT HELLO，V254，D1，S6

也可用 INIT HELLO，V17，D2

或 INIT HELLO

Vv、Ss、Dd 可以任意选用，次序可以互相调换。

其余命令格式在下面各个命令中叙述。

二、DOS 命令的调用

在 BASIC 方式下可以调用 DOS 命令，调用的方式有二种：

（1）立即方式 在英文或中文 BASIC 提示符下，从键盘上直接打入 DOS 命令，DOS 立即解释这个命令。

例如盘上有个打印程序文件，设其程序名为“打印”，如果要装入这个程序，可以在键盘上输入“LOAD 打印”命令。

（2）间接方式 在 BASIC 程序中使用 DOS 命令，方法是把 DOS 命令作为 PRINT 语句中的字符串输出。并在前面加一个特殊标志 CTRL-D 字符，在 APPLE SOFT BASIC 里可以用 CHR$ 函数来定义 CTRL-D，在 ASCⅡ码中，4 代表 CTRL-D 字符，下面的语句定义了一个 CTRL-D 字符并赋值给字符串变量 D$，语句 20 以间接方式执行了 DOS 命令，装入并执行文件名为“打印”的程序。

```
10 D$=CHR$(4)
20 PRINT D$;"RUN 打印"
```

采用这种方式执行 DOS 命令，D$ 之前必须有一个 RETURN 字符，否则 D$ 将不起作用，例如执行如下程序：

```
10 D$=CHR$(4)
20 PRINT "等待";
30 PRINT D$;"RUN 打印"
```

屏幕上显示“等待 RUN 打印”，这是因为语句 20 中的“；”把 RETURN 取消了，为确保程序正确运行，“；”要去掉。

如果 D$ 没有定义或定义错了，仅显示“RUN 打印”而不执行 DOS 命令。

下面的 DOS 命令只能以间接方式调用：

OPEN、APPEND、READ、WRITE、POSITION。

下面的 DOS 命令既可以以立即方式调用，也可以间接方式调用：

CATALUG、SAVE、BSAVE、LOAD、BLOAD、RUN、BRUN、LOCK、UNLOCK、DELETE、RENAME、EXEC、CLOSE、MON、NOMON、PR#、IN#、MAXFILES。

但 DOS 命令中的 INIT 命令只能以立即方式调用，因为 INIT 命令执行后将会把盘片中存放的资料全部清除掉，在程序中使用，后果严重。

三、DOS 命令

DOS 命令必须以大写字母输入。

1．INIT 命令

软盘初始化命令，它把软盘上所有资料清除并把 DOS 抄到 0、1、2 道上，对磁道、扇区

进行规格化。

格式：INIT f[，Vv][，Ss][，Dd]

INIT 命令是唯一一个能指定软盘号码的命令，以后使用这张磁盘时，其它 DOS 命令中的盘片号码必须与它相同，命令才能生效。如果在 INIT 中不给出号码，INIT 命令将自动地设定号码为 254。

在较大内存容量系统中格式化的盘不能在较小内存的系统中执行 DOS 的引导。

2．CATALUG 命令

显示磁盘内的文件目录，如果打印机已接通，打印输出文件目录。

格式：CATALUG

执行这个命令显示 18 个文件目录，按除 RESET、CTRL 及 SHIFT 以外的任何键，继续显示其余的文件目录。

在系统主盘上保存有系统提供的一些程序，比如复制整个盘片的 COPY A 程序，复制正文文件的 FID 程序，以及展示 DOS 功能的 RANDOM 程序等，使用 CATALUG 命令可以读出它们的目录如下：

```
]CATALOG
DISK VOLUME 254
*A 006 HELLO
*I 018 ANIMALS
*T 003 APPLE PROMS
*I 006 APPLE SOFT
*I 026 APPLE VISION
*I 017 BIORHYTHM
*B 010 BOOT13
*A 006 BRIAN'S THEME
*B 003 CHAIN
*I 009 COLOR DEMO
*I 009 COLOR DEMOSOFT
*A 009 COPY
*B 003 COPY OBJO
*A 009 COPYA
*A 010 EXEC DEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
*A 028 LITTLE BRICK OUT
*A 003 MAKE TEXT
*B 009 MASTER CREATE
*B 027 MUFFIN
*A 051 PHONE LIST
*A 010 RANDOM
```

```
*A 013 RENUMBER
*A 039 RENUMBER INSTRUCTIONS
*A 003 RETRIEVE TEXT
 T 004 ANIMALSFILE
 T 004 DO'ER
 A 006 PRIME
 A 002 EVEN MORE RECENT PROGRAM!!
```

目录中每个目占一行，每行从左到右是：

第一项：加锁信息，有*表示这个文件加锁，只能读，不能对它进行修改；

第二项：文件类型码，它的意义见表 4-1-1；

第三项：文件所占的扇区数，用三位数字表示，文件至少占一个扇区，即使空的正文文件也占一个扇区，它存放描述该文件在盘上所占用的磁道及扇区位置的道/扇区表(T/S 表)，对一个非空文件，至少占 2 个扇区，如果一个文件占用的扇区数超过 255 个，目录上显示的数目是扣除了 256 后的长度，这对了解盘空间使用情况有些影响；

第四项：文件名，其规定已述。

在汉字信息处理系统中，其管理程序及数据量都很大，用户可以经常用这条命令了解空间使用情况，当磁盘操作过程中出现"DISK FULL"，可以借助这个命名查阅文件目录，以删去不需要的程序或数据。

汉字系统中可以使用中文文件名，假设盘上有一个打印程序，其文件名为"打印"，另有一个正文文件，如教师名册，其文件名为"名册"。

用这条命令打出目录如下：

```
 ]CATALOG
DISK VOLUME 254
*A 006 HELLO
 T 005 HUN1
 T 013 JS
*T 009 名册
 A 015 ZYI3
 B 034 ZI1
 A 006 打印
```

3．SAVE 命令

这条命令把存放在主存 RAM 中从 $800 开始时 BASIC 用户程序存进磁盘，主机内存中的程序保留。

格式：SAVE f[，Vv][，Ss][，Dd]

例如：主存中现有一个输入汉字报表的程序，要求用文件名"输入"存到盘上，可以使用命令：

SAVE 输入

两个类型相同的程序以同一个文件名存放，后存入的程序将把先存入的程序冲去，使用 SAVE命令要特别注意这一点。如果类型不同，新程序不会被存入磁盘，并产生 "FILE TYPE MISMATCH"的信息。

如果盘片上的缺口已用胶纸贴没，盘片写保护，执行 SAVE 命令得到 “WRITE PROTECTED”信息。

4. LOAD 命令

这条命令从磁盘上读出一个程序文件装入到内存 RAM。

格式：LOAD f[，Vv][，Ss][，Dd]

例如：LOAD 打印

执行这条命令，把原来存在 RAM 中的程序及变量全部清除掉，同时装入这个程序。

设计一个汉字信息处理系统常采用模块结构，使系统便于维护和扩充，为了解决微机内存空间不足的矛盾，系统采用了覆盖技术，这样当两段程序共用变量值时，必须把第一段中使用的变量保存到内存的一个特定区域，第二段程序装入时，尽管第一段的程序及变量值全部被清除，但保存在特定区域的共用变量值没有被清除。这就要求第二段程序执行时，先从这个区域恢复这些变量，然后继续执行，详细介绍见第七节。

如果命令中指出的文件名不在目录表中，显示“FILE TYPE MISMATCH”错误信息。

汉字系统中，正常装入后显示中文 BASIC 提示符，用户可以翻阅修改，或执行这个程序。

5. RUN 命令

装入一个程序文件，并执行这个程序。

格式：RUN f[，Vv][，Ss][，Dd]

例如：RUN 打印，D1

RUN 既可用直接方式调用，也可用间接方式调用，两段程序之间的连接常用间接式的 RUN 命令，它出现在第一段程序的最后。比如第二段程序为打印程序，则命令使用如下：

PRINT CHR$(4);"RUN 打印"

汉字处理系统的程序和数据量往往是很大的，如果把程序和数据分开存放在两个软盘中，程序盘放在 1 号驱动器，数据盘放在 2 号驱动器，在程序中可以交叉用间接方式调用数据及程序文件，只要在命令中写进相应的驱动器号码数就可以实现。

6. LOCK 命令

这条命令使盘片上某一程序得到写保护。

格式：LOCK f[，Vv][，Ss][，Dd]

例如：LOCK 打印，D1

执行这条命令后在打印文件目录第一项出现一个*，表示这个文件写保护，它不会被重写或清除。这样可以防止意外的把它清除或被其它文件覆盖，如果盘片保护缺口封住，LOCK 命令不能执行，当文件写保护时，清除或重写得到“FILE LOCKED”信息。

7. UNLOCK 命令

取消一个文件的写保护。

格式：UNLOCK f[，Vv][，Ss][，Dd]

例如：UNLOCK HELLO

执行这条命令后，目录表中该文件目录行第一项的 * 号消失，可以对这个 HELLO 文件进行修改、重写。例如：

```
]UNLOCK HELLO
]CATALOG
```

```
DISK VOLUME 254
 A 006 HELLO
 T 005 HUN1
 T 013 JS
*T 009 名册
 A 015 ZYI3
 B 034 ZI1
 A 006 打印
```

8．RENAME

更换盘上某个文件的文件名。

格式：RENAME 老名，新名

例如：RENAME JS，教授

执行这个命令，把存放全校教授名册的文件"JS"改名为"教授"。

```
]RENAME HUN3，教授
]CATALOG
DISK VOLUME 254
*A 006 HELLO
 T 005 HUN1
 T 013 教授
*T 009 名册
 A 015 ZYI3
 B 034 ZI1
 A 006 打印
```

这个命令对程序文件、正文文件、二进制文件均起作用。

执行这个命令时，DOS 不检查新名是否在这个盘上已经用过，所以，如果新名已在盘片上，将得到两个名字相同的文件。

若文件写保护，命令不起作用。

9．DELETE 命令

清除盘上某个文件。

格式：DELETE f[，Vv][，Ss][，Dd]

例如：DELETE HUN1

执行这个命令后，盘上 HUN1 文件被清除，目录随之被清除。

```
]DELETE HUN1
]CATALOG
DISK VOLUME 254
*A 006 HELLO
 T 013 教授
*T 009 名册
 A 015 ZYI3
 B 034 ZI1
```

A 006 打印

如果文件写保护，执行 DELETE 命令出现"FILE LOCKED"信息。

10. VERIFY 命令

检查磁盘中某个文件是否完整。

格式：VERIFY f[, Vv][, Ss][, Dd]

例如：VERIFY 打印

由于 DOS 将一个文件写入每一个扇区时，计算存在一个扇区内的字符个数，称为校验和，并存在这个扇区中，当使用 VERIFY 命令时，DOS 重新计算这个校验和，并与存在这个扇区中的校验和比较。如果正确，系统返回 BASIC 状态，如果两个数值不同，显示"I/O ERROR"的错误信息。

11. MAXFILES 命令

DOS 3.3 系统最多同时可以处理 16 个文件，这个命令指定 DOS 可同时处理最多的文件数。

命令格式：MAXFILES n

其中 n 为 1～16 间的整数，如果超过了这个范围，将显示出错信息。

例如：MAXFILES 8 指出一次最多可以同时打开 8 个文件。

DOS 引导后，在 DOS 下方开辟了一个约 1.75 K 的文件缓冲区，DOS 为每个文件规定 595 个字节的缓冲区。每一个缓冲区由各为 256 字节的读、写缓冲区组成，剩下的 83 个字节用来存放有关文件的一些资料，如道 / 区表等，见图 4-3-1。这样的文件缓冲区最多可存放三个文件。DOS 允许通过 MAXFILES 命令来指定更大的文件缓冲区，以使 DOS 最多同时能处理 16 个文件。这些缓冲区开辟在 HIMEM 下面的存贮区域内，n 越大，作为缓冲区的内存空间越大，用户可以使用的存贮空间也就越小，可见增大文件缓冲区是以减少用户可使用内存空间为代价的。

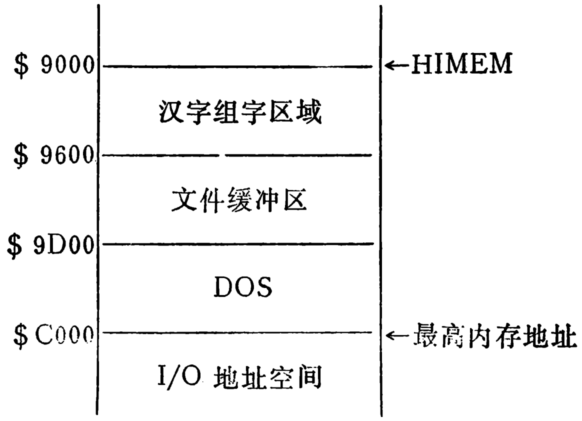


图 4-3-1

如果要使用这条命令，一定要在执行程序或将程序调入内存之前。否则有可能冲掉整数 BASIC 程序或破坏 APPLE SOFT BASIC 中的字符串。

在程序中使用这一命令必须用间接式，而且要放在程序的最前面，否则，它将使 GOTO、GOSUB 及其它指令产生不正常的功能。

10 PRINT CHR$(4);"MAXFILES 8"

MAXFILES 命令以最后一次定义的 n 值生效。

如果打开的文件数已经达到了定义的最大值，不再有可用的文件缓冲区，但使用者又执行了一条 DOS 命令，则显示"NO BUFFERS AVAILABLE"信息。

12. MON 命令及 NOMON 命令

这是一条监督指令，它监督主机与磁盘间交换的信息及磁盘操作命令，通常这些信息及命令是不会显示在屏幕上的，这将给程序的调试带来困难，DOS 设置这条命令可以帮助用户检查程序中的错误。

(1) MON 命令

格式：MON [C][，I][，O]

参数指出所要显示的信息类型。

C　表示监督磁盘操作命令；

I　表示显示从磁盘输入到主机内存的信息；

O　表示显示从主机内存输出到磁盘的信息。

这些参数可以依照任何顺序排列或任何组合方式使用，不过最少要有一个参数出现在命令中，否则这个命令将不被执行。三个参数组合成下面七种可能的监督形式：

命　令	监　督　功　能
MON C	监督磁盘操作命令
MON I	监督从磁盘输入到主机内存的信息
MON O	监督从主机内存输出到磁盘的信息
MON C，I	同时监督磁盘操作命令及磁盘输入到内存的信息
MON C，O	同时监督磁盘操作命令及内存输出到磁盘的信息
MON O，I	同时监督磁盘输入到内存及内存输出到磁盘的信息
MON C,I,O	同时监督磁盘操作命令，磁盘输入到内存以及内存输出到磁盘的信息。

(2) NOMON 命令

NOMON 命令是专为取消 MON 命令而设置的，

格式：NOMON[C][，I][，O]

参数的含义及用法和 MON 命令中的规定相同。

MON 命令执行后，其功能一直有效，直到执行了 NOMON，INT 或 FP 命令为止，重新启动，按 RESET 键或重新加载时，MON 命令也被取消。

如果执行了 MON I，O，C 命令后又输入 NOMON O，将使内存输出到磁盘的信息不受监督，也即不显示出来，至于由磁盘输入到内存的信息及 DOS 命令仍然显示在屏幕上。

第四节　中文自动启动程序

第二章中曾说过每次由英文 BASIC 进入中文 BASIC 都必须重复输入 PR#CN 命令，如果把下面的中文自动启动程序加在用户程序的最前面，就可以在英文 BASIC 下直接装入执行汉字信息处理程序，或在开机启动后使 APPLE Ⅱ 直接进入汉字系统。

中文自动启动程序清单

```
10 HIMEM:36864
20 POKE 37984,0
30 PR#3
```

```
 40 PRINT "␣"
 50 POKE 43603, 3
 60 POKE 43604, 192+3
 70 POKE 43605, 48
 80 POKE 43606, 192+3
 90 POKE 54, 189
100 POKE 55, 158
110 POKE 56, 129
120 POKE 57, 158
130 FOR I=1 TO 15
140 GET A$
150 NEXT I
```

一、由英文 BASIC 直接进入中文 BASIC

具体步骤如下：

（1）在英文 BASIC 下输入PR#CN 命令使机器进入中文 BASIC；

（2）在汉字系统下输入上述中文自动启动程序，并从 160 语句开始输入汉字信息处理的用户程序；

（3）输入"SAVE A"命令，A 为用户程序文件名，把整个程序作为一个用户文件存放到磁盘上。

以后每次要使用这个用户程序时，只要在英文 BASIC 下输入"RUN A"命令，机器就会自动进入中文 BASIC，屏幕上先出现第二章第一节所示的字幕，紧接着自动转去执行用户程序。

二、主机启动后直接进入中文 APPLE SOFT BASIC

由于 APPLE Ⅱ PLUS 微机上配有"自动启动 ROM"，在初始开机通电即所谓冷启动时，如果系统配有软盘驱动器，并且在其中放入了经过初始化的盘片，那未监控系统会自动启动软盘驱动器，进入盘片上的招呼程序（HELLO 程序），所以，我们可以把中文自动启动程序作为 HELLO 程序，这样，每次只要把盘片放进 1号软盘机，再打开电源，就会自动装入并执行中文自动启动程序。直接进入中文 BASIC。

如果用户希望在主机启动后直接进入中文信息处理的某一用户程序，可以采用下面两种方法：

（1）把中文自动启动程序放在用户程序的最前面，然后把整个程序作为 HELLO 程序，这样每次主机启动后直接进入中文 BASIC，并且紧接着执行用户程序。

（2）在中文自动启动程序后面加一条调用用户程序的 DOS 命令，再把整个程序作为 HELLO 程序。

具体步骤如下：

（1）把 DOS 3.3 盘片放进 1 号软盘器，启动主机；

（2）输入 NEW 命令；

（3）输入 INIT HELLO 命令对新盘片进行初始化，或输入 PR#CN 后，使用 INIT

HELLO 命令初始化盘片；

（4）输入汉字处理的用户程序，假设用户程序文件名为“HZ”，使用“SAVE HZ”命令把用户程序存贮到软盘上；

（5）输入 NEW 命令，清除用户程序；

（6）输入中文自动启动程序，接着再输入一个语句 160 PRINT CHR$(4);″RUN HZ″；

（7）输入 SAVE HELLO 命令，整个程序作为 HELLO 程序存贮在软盘上。这一步可以在英文 BASIC 下完成，但如果使用汉字作为用户程序文件名，这一步只能在中文 BASIC 下完成。

以后每次使用这个用户程序时，只要把这个盘片放入 1 号软盘器，再打开电源，就会自动进入汉字系统，显示上面所述的字幕，由于 DOS 装入到主存，所以执行 160 这一句时，DOS 操作系统把软盘上的用户程序 HZ 装入到主存，并执行这个用户程序。

这二种方法给用户使用程序提供了方便。

例如：下面的程序作为 HELLO 程序，开机后进入汉字系统，先显示“汉字报表处理程序”的中文字幕，并装入用户程序（程序文件名为 HZ），这时屏幕上显示的中文字幕一直停留到用户程序装入完成。

```
 10 HIMEM:36864
 20 POKE 37984，0
 30 PR#3
 40 PRINT ″␣″
 50 POKE 43603，3
 60 POKE 43604，192+3
 70 POKE 43605，48
 80 POKE 43606，192+3
 90 POKE 54，189
100 POKE 55，158
110 POKE 56，129
120 POKE 57，158
130 FOR I=1 TO 15
140 ET A$
150 NEXT I
160 CALL 49941
170 POKE 215，4:POKE 214，10:PRINT ″汉字报表处理程序″
180 CALL 49944
190 D$=CHR$(4):PRINT D$;″RUN HZ″
```

第五节 正 文 文 件

汉字信息处理系统中，数据在整个系统中占有重要的地位，它不仅信息量大，而且经常反复被使用，DOS 操作系统建立了一个文本文件（常称正文文件）来保存这些信息。正文文件有两种，一种是顺序文件，另一种是随机文件。在磁盘目录索引中，它们都以字母 T 作为标

识，顺序文件和随机文件的读写方式不同，顺序文件一般以顺序方法读写，随机文件可作随机存取，即可以从任一指定的记录处开始读写，所以随机文件用于须快速存取或数据要经常变动的处理过程。

正文文件不同于程序文件，它不能用 LOAD、SAVE 命令存取，DOS 操作系统为正文文件规定了读写步骤：读写之前必须先打开文件，读写之后要关闭文件，并提供了一套读写命令，它们是：

OPEN、CLOSE、READ、WRITE、APPEND、POSITION、EXEC。

其中命令 OPEN、APPEND、READ、WRITE、POSITION只能用间接方式调用，CLOSE、EXEC 可用立即方式调用。

除了这些命令外，其它的 DOS 命令还有：

LOCK、UNLOCK、DELETE、RENAME、MON、NOMON、VERIFY、CATALOG。

在正文文件上和在程序文件上一样使用。

OPEN、CLOSE、READ、WRITE 及 POSITION 命令在顺序文件和随机文件中均可使用，但用法略有不同。

顺序文件和随机文件都以记录方式存贮信息，每个记录间用 RETURN 分隔，顺序文件中记录是不定长的，随机文件中记录是定长的，信息均以 ASC Ⅱ 码方式存贮。

一，顺序文件

顺序文件的每个记录可以由 1～32767 个字符组成，每个字符都以 ASC Ⅱ 码方式存贮，各个记录间用 RETURN 分隔。

例如："APPLE"、"TEXT"、"FILE"作为三个记录，在顺序文件中存贮形式如图4-5-1。

字符	A P P L E ↲ T E X T ↲ F I L E ↲
ASC Ⅱ 码	65 80 80 76 69 13 84 69 88 84 13 70 73 76 69 13 00 00…00

图 4-5-1

中文信息以中文字母的 ASC Ⅱ 码方式存贮。

例如："部门"，"姓名"作为两个记录，存贮形式如图 4-5-2。

中文字母	卜 口 弓 中 日 弓 ↲ 女 竹 平 一 弓 戈口 ↲
ASC Ⅱ 码	79 72 6E 6C 32 61 6E 32 13 76 68 71 6D 32 6E 69 72 32 13 00

图 4-5-2

顺序文件每个记录可以有不同的长度，如"姓名"占 10 个字节，"部门"占 9 个字节(其中 RETURN 占一个字节)，记录与记录之间不留空隙。

1. 顺序文件的打开与关闭

打开、关闭顺序文件的命令是 OPEN 和 CLOSE 命令。

(1) OPEN 命令

格式：OPEN　f[，Vv][，Ss][，Dd]

OPEN 命令打开一个顺序文件，如果是个新文件，则这个命令建立一个新文件，如果文件已经被打开，则命令先把这个文件关闭，再打开，并把文件的定位指针指向这个文件的首字节，同时 OPEN 命令在内存的文件缓冲区中为文件 f 建立 595 字节的工作区（如果 MAXFILES 命令指定的文件个数大于 3 个，这个工作区可能建在内存中汉字组字区域的下面），并指定以后读、写命令所需要的扩充插座号码及驱动器号码。

（2）CLOSE 命令

格式：CLOSE [f]

CLOSE 命令关闭一个文件，并把 OPEN 命令申请的工作区归还主机，如果没有指定 f，DOS 关闭所有处于打开状态的文件，除了被 EXEC 命令支配的文件。

2．顺序文件的读写命令

对已用 OPEN 命令打开的文件，可用 WRITE 和 READ 命令进行读写操作。

（1）WRITE 命令

格式：WRITE f[，Bb]

WRITE 命令指定文件的写操作，写入文件 f 的所有信息在 PRINT 语句中指明，DOS 把 PRINT 语句中的信息写到 OPEN 命令申请的工作区，然后再从工作区写到盘上，工作区和磁盘之间的信息交换以页交换方式进行(一页为 256 个字节)。当工作区中写满 256 个字节，DOS 将这一页写入盘内未经分配的扇区，如果工作区中未写满 256 个字节，DOS 读到 CLOSE 命令，DOS 将信息写入一个扇区，并以 0 补足256 个字节。汉字信息直接写在 PRINT 语句中。

顺序文件写入过程中，文件的定位指针依序递增，写入的信息不在屏幕上显示。

B 参数表示从文件的第 b 字节开始写入(首字节即第 0 字节)。

WRITE 功能在任一 DOS 命令或 INPUT 语句执行时失效。

WRITE 命令只能以间接方式调用。

（2）READ 命令

格式：READ f[，Bb]

READ 命令定义读文件操作，从文件 f 中读取信息用 INPUT 语句。执行 INPUT 语句，DOS 从磁盘上指定的文件 f 中读一个记录到工作缓冲区，再把信息指定给 INPUT 语句中的变量。

B 参数指定从文件的第 b 字节开始读取，READ 命令也只能以间接方式调用。

3．顺序文件的写入和读出

（1）顺序文件的写入，需要顺序执行以下的命令序列：

OPEN　　打开文件

WRITE　　指定写操作

PRINT ┐
⋮　　 ├ PRINT 语句中包含要写入的信息
PRINT ┘

CLOSE 关闭文件

例 1　建立文件 EXA1，并写入字符串“APPLE”，“TEXT”，“FILE”

```
10 REM EXA1
20 D$ = CHR$( 4)
30 PRINT D$;"OPEN EXA1"
40 PRINT D$;"WRITE EXA1"
50 PRINT "APPLE"
60 PRINT "TEXT"
70 PRINT "FILE"
80 PRINT D$;"CLOSE EXA1"
90 END
```

文件 EXA1 中有三个记录，其存贮形式如图 4-5-1 所示，RETURN 字符插在每个 PRINT 语句中字符的后面。

如果需要指定扩充插座号码 Ss，驱动器号码 Dd，及盘片卷号Vv参数，只要在第一个DOS 命令，即 OPEN 命令里指出，这些参数必须写在双撇号内，语句 30 修改如下：

```
30 PRINT D$;"OPEN EXA1, V123, S6, D1"
```

若在标号 50，60 的 PRINT语句后面使用分号，如下程序所示：

```
10 REM EXA2
20 D$ = CHR$(4)
30 PRINT D$;"OPEN EXA2"
40 PRINT D$;"WRITE EXA2"
50 PRINT"APPLE";
60 PRINT"TEXT";
70 PRINT"FILE"
80 PRINT D$;"CLOSE EXA2"
90 END
```

则文件 EXA2 只有一个记录，记录中存放了 3 个字符串串接所形成的新字符串，其存贮形式如图 4-5-3 所示。这是因为在 PRINT 语句中后面出现分号时不插入 RETURN 字符。

A P P L E T E X T F I L E ↵

65 80 80 76 69 84 69 88 84 70 73 76 69 13 00 00 00 00

图 4-5-3

(2) 顺序文件的读出需要顺序执行以下的命令：

OPEN 打开文件

READ 指定读操作

INPUT
⋮ 从盘上读出的信息输入到变量或数组中
INPUT

CLOSE 关闭文件

例 2　试读取例 1 所建立的 EXA 1 文件。

```
100 REM READ FILE EXA1
110 D$ = CHR$(4)
120 PRINT D$;"OPEN EXA1"
130 PRINT D$;"READ EXA1"
140 INPUT A$
150 INPUT B$
160 INPUT C$
170 PRINT D$;"CLOSE EXA1"
180 END
```

字符串"APPLE"、"TEXT"、"FILE"分别读到字符串变量 A$、B$、C$ 中。

读取EXA2文件的程序如下：

```
110 REM READ FILE EXA2
120 D$ = CHR$(4):PRINT D$;"OPEN EXA2"
130 PRINT D$;"READ EXA2"
140 INPUT A$
150 PRINT D$;"CLOSE EXA2"
160 END
```

A$ 中为字符串"APPLE TEXT FILE"

如果用读取文件 EXA1 的程序读取文件 EXA2，将显示错误信息：

```
END OF DATA
BREAK IN 150
```

并中止执行读程序，这是因为语句 140 读出 EXA2 仅有的一个记录，所以执行语句150时，没有数据可读了。

但用读取 EXA 2 的程序可以读 EXA 1 文件，这时 A$ 中为 "APPLE"，屏幕上显示"EXTRA"，表示文件中还有数据。

上述读写程序执行时，DOS 命令、输入磁盘的数据及从盘上读取出的数据都不会显示在屏幕上，为了监视读写过程，可以使用监督命令 MON C，I，O。

例如：在例 1 程序执行之前，打入 MON C，I，O 命令，然后用 RUN 命令启动执行，屏幕上将显示：

```
OPEN EXA1
WRITE EXA1
APPLE
TEXT
FILE
CLOSE EXA1
```

然后输入 NOMON C，I，O 命令撤消监督。

也可以在程序中使用 MON 命令，如把例 2 程序修改如下：

```
100 REM READ FILE EXA1
110 D$ = CHR$(4):PRINT D$;"MON C, I, O"
```

```
120 PRINT D$;"OPEN EXA1"
130 PRINT D$;"READ EXA1"
140 INPUT A$
150 INPUT B$
160 INPUT C$
170 PRINT D$;"CLOSE EXA1"
180 PRINT D$;"NOMON C, I, O"
190 END
```

执行这个程序，屏幕上将显示：

```
OPEN EXA1
READ EXA1
?APPLE
?TEXT
?FILE
CLOSE EXA1
NOMON C, I, O
```

通常读、写程序中，INPUT 语句和 PRINT 语句中的变量个数必须相同，变量类型必须一一对应，并按相同的顺序读写。

例如：建立 EXA3 文件的程序如下：

```
10 INPUT A$ :INPUT B$:INPUT C
20 D$=CHR$(4):PRINT D$;"OPEN EXA3"
30 PRINT D$;"WRITE EXA3"
40 PRINT A$:PRINT B$:PRINT C
50 PRINT D$;"CLOSE EXA3"
60 END
```

读 EXA3 文件的程序如下：

```
100 PRINT D$;"OPEN EXA3"
110 PRINT D$;"READ EXA3"
120 INPUT A$:INPUT B$:INPUT C
130 PRINT D$;"CLOSE EXA3"
140 PRINT A$:PRINT B$:PRINT C
150 END
```

逗号不能出现在要写入的字符串中。

上例如输入"APPLE，TEXE"到 A$中，则机器将显示信息：

? EXTRA IGNOROD(但不影响 B$、C 的输入)。

当读文件 EXA3 时，A$ 中仅有 "APPLE"，即"，"及它的后面信息丢失了，但 B$、C 中仍有原输入的信息。

4．中文信息的读写

中文信息是作为字符串存贮在顺序文件中。

例 3 输入一个报表的表头、副表头、报表栏标题及各行，并建立一个顺序文件，文件

名可自由选取。

```
  5 REM **输入报表**
 10 CALL 49941:E$=CHR$(4)
 20 INPUT "表头";A$
 30 INPUT "副表头";B$
 40 INPUT "栏数，行数:";N, M: DIM C$(N):DIM D$(M, N)
 50 FOR I=1 TO N
 60 PRINT "第";I;"栏标题";
 70 INPUT "␣";C$(I)
 80 NEXT I
 90 FOR I=1 TO M
100 PRINT "第";I;"行"
110 FOR J=1 TO N
120 PRINT C$(J);
130 INPUT"";D$(I, J)
140 NEXT J
150 NEXT I
200 REM **建立报表文件**
210 INPUT "文件名:";F$
300 PRINT E$;"OPEN";F$
330 PRINT E$;"WRITE";F$
340 PRINT A$;PRINT B$:PRINT N:PRINT M
350 FOR I=1 TO N:PRINT C$(I):NEXT I
360 FOR I=1 TO M:FOR J=1 TO N:PRINT D$(I, J):NEXT J:NEXT I
370 PRINT E$;"CLOSE";F$
380 END
```

报表栏数N及行数M，必须写入文件，以便在读取报表各栏标题以及各报表行时作为循环变量的终值。

```
400 REM **读报表文件**
410 PRINT E$;"OPEN";F$
420 PRINT E$;"READ";F$
430 INPUT A$, B$, N, M
440 FOR I=1 TO N:INPUT C$(I):NEXT I
450 FOR I=1 TO M:FOR J=1 TO N:INPUT D$(I, J):NEXT J:NEXT I
460 PRINT E$;"CLOSE";F$
480 END
```

若要监视读写过程，可加入语句：

```
290 PRINT E$;"MON C, I, O"
470 PRINT E$;"NOMON C, I, O"
```

若允许用户选择参数 Vv、Ss 及 Dd，可加入语句：

220 INPUT "扩充插座号："; S

230 INPUT "驱动器号："; D

240 INPUT "盘片号："; V

打开文件时，必须指定这些参数，即语句300、410修改如下：

300 PRINT E$;"OPEN";F$;", S";S;", D";D;", V";V

410 PRINT E$;"OPEN";F$;", S";S;", D";D;",V";V

如果用户需要修改文件，修改后重写入时会把存在文件中原来的信息覆盖掉，所以写入的信息至少应当与原文件中的信息一样长，否则重写后，文件后面会遗留原文件的信息。

例如：原文件中有两个记录"姓名"，"出生年月"，若修改为"姓名"，"年龄"后重写，结果如下（为方便用英文形码表示）：

原文件　VHQM␣ NIR␣ ↲UU␣ HQM␣ OQ␣ B␣↲

写入内容　VHQM␣NIR␣ ↲OQ␣ OINI␣↲

结果文件　VHQM␣ NIR␣ ↲OQ␣ OINI␣↲␣B␣↲

为了消除后面的无用信息"␣B␣↲"，可在存入新的信息之前先将原有信息清除掉，在OPEN 命令之前使用 DELETE 命令可以达到这个目的，但是，对一个新建立的文件执行 DELETE 命令，由于文件不存在，将得到"FILE NOT FOUND"的错误信息。为了保证程序能正确执行，必须在 DELETE 命令之前另外加一个 OPEN 命令。

300 PRINT E$;"OPEN";F$;", S";S;", D";D;", V";V

310 PRINT E$;"DELETE";F$

320 PRINT E$;"OPEN";F$

第一个 OPEN 命令当 F$ 不存在时，建立这个文件。

不论 F$ 文件何时被建立，DELETE 命令都能把这个文件清除掉。

第二个 OPEN 命令将建立一个空文件。

5．APPEND 命令

使用 DOS 提供的 APPEND 命令，可以在顺序文件中增加信息。

把一个报表写入顺序文件并关闭这个文件后，若要再次打开它，OPEN 命令把文件的定位指针指向文件的首字节。如果我们要在这个文件的后面增加一些报表项，可以按顺序把整个报表都读出来，然后再扩充，但是，对于信息量大的文件需要花费很多时间，且受主机内存容量的限制。APPEND 命令可以把文件的定位指针指向文件的最后一个字节的下一个位置，这样可方便地进行扩充。

格式：APPEND f[, S_s][, V_v][, D_d]

APPEND 命令打开一个文件，并使文件的定位指针指向文件的最后一个字节的下一个位置。

APPEND 不能象 OPEN 命令那样建立一个新文件，如果 APPEND 要打开的文件不存在，将得到"FILE NOT FOUND"的错误信息。

APPEND 命令后只能跟 WRITE 命令，不能跟 READ 命令，如果用了READ 命令，会出现错误信息"END OF DATA"。

例4　建立一个顺序文件，写入学生成绩表。

成绩表格以 DATA 语句建立，包括栏标题，并以空格作为表格结束标志，程序如下：

10 INPUT "文件名："; F$

```
 20 E$=CHR$(4):PRINT E$;"OPEN";F$
 30 PRINT E$;"DELETE";F$
 40 PRINT E$;"OPEN";F$
 50 READ A$, B$, C$
 60 IFA$="␣"THEN 90
 70 PRINT E$;"WRITE";F$
 80 PRINT A$:PRINT B$:PRINT C$:GOTO 50
 90 PRINT E$;"CLOSE";F$
100 DATA "姓名", "学号", "成绩"
110 DATA "王林", "123512", "85"
120 DATA "陈红", "123514", "90"
130 DATA "李明", "123516", "75"
140 DATA "␣", "␣", "␣"
150 END
```

在原有顺序文件之后，写入新数据，新数据以 DATA 语句建立，程序如下：

```
 10 INPUT "文件名:"; F$
 20 E$=CHR$(4):PRINT E$;"APPEND";F$
 30 READ A$, B$, C$
 40 IF A$="␣" THEN 70
 50 PRINT E$;"WRITE";F$
 60 PRINTA$:PRINT B$:PRINT C$:GOTO 30
 70 PRINT E$;"CLOSE";F$
 80 DATA "张玲","123518", "82"
 90 DATA "林青", "123520", "92"
100 DATA "␣", "␣", "␣"
110 END
```

例 5 上例中如果数据以 INPUT 语句输入，并允许用户在任意一行输入结束后中止输入。

程序如下：

```
 10 INPUT "文件名:"; F$
 20 E$=CHR$(4):PRINT E$;"OPEN";F$
 30 PRINT E$;"DELETE";F$
 40 PRINT E$;"OPEN";F$
 50 PRINT "输入姓名，学号，成绩:";
 60 INPUT "␣";A$, B$, C$
 70  PRINT E$;"WRITE";F$
 80 PRINT A$:PRINT B$:PRINT C$
 90 PRINT E$
100 PRINT "继续输入? 0:结束"
110 GET U$:IF U$< > "0" THEN 50
```

```
120 PRINT:PRINT E$;"CLOSE";F$
130 END
```

值得注意的是，执行语句100时，将输出提示信息。在WRITE命令执行之后，它将被存到文件中。为了避免这一现象，在每次写入 A$、B$、C$ 之后，必须使得 WRITE 命令失效，任何 DOS 命令都能达到这一目的，语句 90 就是为此目的而设置的。

本程序允许用户在任意一行输入后中断输入。

若要在文件中加入新数据，程序如下：

```
 10 INPUT "文件名:"; F$
 20 E$=CHR$(4):PRINT E$;"APPEND";F$
 30 PRINT "输入姓名、学号、成绩:";
 40 INPUT "␣";A$, B$, C$
 50 PRINT E$;"WRITE";F$
 60 PRINT A$;PRINT B$:PRINT C$
 70 PRINT E$
 80 PRINT "继续输入? 0:结束"
 90 GET U$:IF U$<> "0" THEN 30
100 PRINT:PRINT E$;"CLOSE";F$
110 END
```

用下面的程序读出这个文件，你可以看到新数据已加在原文件的最后，假设一共写入 6 行(包括标题)。

```
10 INPUT "文件名:"; F$
20 E$=CHR$(4):PRINT E$;"OPEN";F$
30 PRINT E$;"READ";F$
40 FOR I=1 TO 6
50 INPUT A$, B$, C$
60 PRINT A$;"␣";B$;"␣";C$
70 NEXT I
80 PRINT E$;"CLOSE";F$
90 END
```

5. POSITION 命令

利用 POSITION 命令可以读出或写入顺序文件中任一段记录的信息。

格式：POSITION f[, Rp]

其中 Rp 表示需要读、写的记录在文件中与当前记录的相对位置，如果 P=0，则从当前所在的记录往前读或写；如果 P=1，则从当前记录的下一个记录开始读或写，如此类推。这个命令的功能是把记录的指针从当前所指的记录位置向前移 P个位置。

使用 POSITION 命令之前，文件 f 必须被打开，由于 OPEN 命令把记录指针指向首记录(P=0)，若在 OPEN 命令后紧接着使用 POSITION 命令，P 就是记录的绝对位置，若 POSITION 命令使指针指向最后一个记录以外的位置，将产生"END OF DATA"的错误信息。

如同其它的 DOS 命令一样，POSITION 命令会使 READ 或 WRITE 命令失效，所以

POSITION 命令必须用在 READ 或 WRITE 命令之前。

例 6　用如下的程序读出例 4 中建立的文件，试给出屏幕上显示的信息，假设文件名为“成绩表”。

```
 10 INPUT "文件名:"; F$
 20 E$ = CHR$(4)
 30 PRINT E$;"MON C, I, O"
 40 PRINT E$;"OPEN";F$
 50 PRINT E$;"POSITION";F$;", R4"
 60 PRINT E$;"READ";F$
 70 INPUT A$
 80 PRINT E$;"POSITION";F$;", R1"
 90 PRINT E$;"READ";F$
100 INPUT B$
110 PRINT E$;"OPEN";F$
120 PRINT E$;"POSITION";F$;", R1"
130 PRINT E$;"READ";F$
140 INPUT C$
150 INPUT D$
160 PRINT E$;"CLOSE";F$
170 PRINT E$;"NOMON C, I, O"
180 END
```

执行此程序，屏幕上将显示：

```
OPEN　成绩表
POSITION　成绩表R4
READ　成绩表
?123512
POSITION　成绩表R1
READ　成绩表
?陈红
OPEN　成绩表
POSITION　成绩表R7
READ　成绩表
?123514
?90
CLOSE　成绩表
NOMON C, I, O
```

利用 POSITION 命令进行修改及重写操作，要特别注意输入的新记录中必须和文件中的原记录长度一致，若新记录比原记录长，则新记录会侵入下一个记录，从而使下一个记录受到破坏，若新记录比原记录短，如例 6 中所示，将会多出一个记录，使下面的记录全部错位。

二、随机文件

顺序文件能有效地利用盘空间，但由于记录不是定长的，各个记录是顺序存放，在数据维护时，譬如修改、插入、删除操作等，需要把数据顺序读出作相应的维护后再重新写入。在检索时，必须把数据一个个读出再作比较，这对于一个信息量较大的汉字处理系统来说，要花费较多的时间，而随机文件的建立相当有规则，存取或修改时，相当快速简易，它适用于需快速存取某一记录或数据经常变动的场合。

随机文件和顺序文件一样由记录组成，然而随机文件各记录间是定长的，当新建一个随机文件时，必须先指定记录长度，以后每写入一个记录，都以这个记录为标准，记录的长度以字节为单位，它是根据文件中最长的记录长度来定义的，一个记录的最大长度取决于组成这个记录的字符个数(包括 CR 及逗号)，一个汉字字符串的长度包括编码的中文字母，组字的空格符及 CR 的总个数，由于一个汉字编码取码为 1～5 码不等，所以对汉字字符串记录定义文件的记录长度时，要特别注意这一点。

在随机文件中信息以 ASC Ⅱ 码存放，当一个记录的实际长度比定义的记录长度短时，不足部分用 0 补上。

随机文件的一个记录中可以包含几个项，每项之间以 RETURN 分隔，如下面的记录有三个项，记录长度定义为 17 个字节。

```
      |A P P L E ↵ T E X T ↵ F I L E ↵                    |

      |65 80 80 76 69 13 84 69 88 84 13 70 73 76 69 13 00 |
       └──────────────┘ └───────────┘ └──────────────┘
项            1              2               3
       └──────────────────────────────────────────┘
记录                         1
```

"APPLE"，"TEXT"，"FILE" 三个记录在随机文件中存贮形式如下：(其中定义记录长度为 7 个字节)

```
字符      |A P P L E ↵      T E X T ↵      F I L E ↵             |

ASC Ⅱ    |65 80 80 76 69 13 00 84 69 88 84 13 00 00 70 73 76 69 13 00 00|

记录字节   0 1 2 3 4 5 6   0 1 2 3 4 5 6   0 1 2 3 4 5 6
          └─────────────┘ └─────────────┘ └─────────────┘
记录             0               1               2
```

中文信息以中文字母的 ASC Ⅱ 的方式存贮。

"部门"、"姓名"二个记录在随机文件中存贮形式如下：(定义记录长度为 11 个字节)

```
中文字母   |卜 口 弓 中 ␣ 日 弓 ␣ ↵      女 竹 平 一 ␣ 弓 戈 口 ␣ ↵|

ASC Ⅱ 码  |79 72 6E 6C 32 61 6E 32 13 0000 76 68 71 6D 32 6E 69 72 321300|
```

随机文件每个记录有固定的长度，DOS 利用这个长度来确定每一个记录第一个字节的

位置，因而能更迅速、更方便地从任一指定的记录处进行读写，提高数据维护及检索的处理速度。

1．随机文件的打开和关闭

随机文件的建立和存取方式与顺序文件极为相同，打开及关闭文件的命令也是 OPEN 和 CLOSE 命令，其中 CLOSE 命令和顺序文件中的格式及用法完全相同，但随机文件使用的 OPEN 命令多了一个记录长度参数 L，它必须被定义。

OPEN 命令

格式：OPEN f，Lj[，Ss][，Dd][，Vv]

其中，L 参数设定记录长度，j 定义一个记录最多能占的字节数，j 的范围是 1～32767，如果省略 j，则隐含 j 为 1，其它参数见 DOS 命令的格式说明。

和在顺序文件中使用的 OPEN 命令一样，这个命令也为随机文件 f 申请一个 595 字节的文件缓冲区，并定义记录长度为 j 个字节。

2．随机文件的读写命令

随机文件的读写命令 WRITE 及 READ 都比顺序文件中使用的相应命令多了一个参数 R。WRITE 及 READ 命令格式如下：

WRITE f[，Rr][，Bb]

READ f[，Rr][，Bb]

其中，R_r 指定记录号，首记录 r＝0，然后依次递增，r 省略，隐含 r 为 0；

B_b 指定一个记录中第 b 个字节，b 省略，隐含 b 为 0，为首字节。

写入操作时，每个记录按照 OPEN 命令定义的记录长度 Lj 使用盘空间，记录是 DOS 处理的单位，只要一个记录的信息长度在 L_j 定义的范围内，任何写入或重写都不会影响到其它记录，但绝不能将一个超过定义的记录长度的信息写入到一个记录空间，尽管这些信息能正确写入磁盘，但当开始写下一个记录时，DOS 仍以 Lj 计算这个新记录的开始位置，因此新记录会覆盖掉前面过长记录的最后部分字符，包括最后的结束记号 RETURN 字符，造成信息混乱，在汉字信息处理系统中使用随机文件，必须予以重视，记录长度的选择要恰当，既不过多地浪费盘的空间，又要防止出现上述现象，随机文件的写入过程与顺序文件相同，不再赘述。

读操作时，OPEN 命令中的 Lj 参数应与建立这个文件时 OPEN 命令中的 Lj 参数相同，如果不相同，则 DOS 按照新定义的 Lj 参数来计算每一个记录的长度及位置。因为读出的信息混乱不堪，通常在建立一个随机文件之后，须将有关记录结构的信息如记录长度及内容作为索引备查，比如存到文件的首记录中，将长度包含在文件名中也不失为一个好办法。如 FILE-L100，名册-L100。

READ 和 WRITE 命令中的 R 参数用来指定从文件中的某一记录开始读写，DOS 利用 Lj 来计算前面所有记录所占的字节数，这个字节数除以 256 所得的值指出了要找的记录在文件中的第几个区段，然后 DOS 查阅每个文件在盘上存放位置的道/区表，以找到这个记录在盘上存储空间的道号、扇区号，DOS 跳过前面的各记录，直接到相应道相应扇区中读取信息或把信息写入，那些被跳过的记录仍保持不变。

WRITE 和 READ 命令中除了 R 参数之外尚有 B 参数，它表示从所指定的记录中特定的字节开始读写，这个字节可以在目前记录中的位置之前或之后。为了有效地使用这个参数，必须对所指定的记录中每一字节的内容了如指掌，特别是包含有汉字信息的记录，不仅要对

每个中文字的中文编码一清二楚，而且不能忘记每个中文字存贮时包括一个组字的空格符，用户往往容易忽视这一点。

3．随机文件的写入和读出

(1) 写入一个特定的记录

例 1 输入一个学生的“学号”、“姓名”、“成绩”并将它们写入指定的文件。

程序如下：

```
10 D$=CHR$(4)
20 INPUT "文件名："; F$
30 INPUT "学号："; A$
40 INPUT "姓名："; B$
50 INPUT "成绩："; C$
60 PRINT D$;"OPEN";F$;", L60"
70 PRINT D$;"WRITE";F$;", R1"
80 PRINT A$:PRINT B$:PRINT C$
90 PRINT D$;"CLOSE"
```

语句 80 把一个学生的学号、姓名、成绩写入指定文件的第二个记录中，这个文件中仅有一个记录，一个记录中有三个项，项与项之间用 RETURN 分隔。

读这个文件的程序如下：

```
10 D$=CHR$(4)
20 INPUT "文件名："; F$
30 PRINT D$;"OPEN";F$;",L60"
40 PRINT D$;"READ";F$;", R1"
50 INPUT A$, B$, C$
60 PRINT D$;"CLOSE"
```

用下列程序读出的结果相同：

```
10 D$=CHR$(4)
20 INPUT "文件名";F$
30 PRINT D$;"OPEN";F$;", L60"
40 PRINT D$;"READ";F$;", R1"
50 INPUT A$:INPUT B$:INPUT C$
60 PRINT D$;"CLOSE"
```

(2) 写入多个记录

例 2 试从一个随机文件的第二个记录开始写入 3 个学生的成绩。

```
10 D$=CHR$(4)
20 OP$=D$+"OPEN":CL$=D$+"CLOSE":WR$=D$+"WRITE":
   DE$=D$+"DELET"
30 INPUT "文件名："; F$
40 PRINT OP$;F$
50 PRINT DE$;F$
60 PRINT OP$;F$;", L60"
```

```
70 FOR R = 1 TO 3
80 INPUT "学号，姓名，成绩："; A$, B$, C$
90 PRINT WR$;F$;", R"; R
100 PRINT A$:PRINT B$:PRINT C$
110 NEXT R
120 PRINT CL$;F$
130 END
```

文件 F$ 中每个记录长度都是 60 个字节，循环语句写入 3 个记录。

读取这个文件的程序如下：

```
10 INPUT "文件名："; F$
20 D$ = CHR$(4):PRINT D$;"OPEN";F$;", L60"
30 FOR R = 1  TO 3
40 PRINT D$;"READ";F$;", R";R
50 INPUT A$, B$, C$
60 PRINT A$;"␣␣";B$;"␣␣";C$
70 NEXT R
80 PRINT D$;"CLOSE";F$
```

4．随机文件应用实例

下面给出随机文件应用的一个实例。

例 3 建立一个随机文件，写入学生成绩表，并根据选择的行号显示学生成绩或修改表格中各项的内容。成绩表格以 DATA 语句给出（包括栏标题），并以空格作为表格结束标志。

可以先建立一个空文件：

```
10 INPUT "文件名：";F$
20 D$ = CHR$(4):PRINT D$;"OPEN";F$;", L60"
30 PRINT D$;"WRITE";F$;", R0"
40 PRINT 0
50 PRINT D$;"CLOSE";F$
```

其中语句 20 建立一个随机文件，定义记录长度为 30 个字节。语句 30 指定把数据写到文件的首记录中，语句 40 在首记录中写入数字 0，以后用首记录存该文件中最后一个记录号。

下面的程序允许用户写入成绩表，根据用户选择的行号显示整个成绩表或某一个学生的成绩以及修改文件中所有的项或某些项。

```
10 D$ = CHR$(4)
20 OP$ = D$ + "OPEN":CL$ = D$ + "CLOSE":RD$ = D$
 + "READ":WR$ = D$ + "WRITE"
30 INPUT F$
40 PRINT OP$;F$;", L60"
50 PRINT RD$;F$;", R0"
60 INPUT M
```

```
70 PRINT D$
80 CALL 49941
90 POKE 215, 1:POKE 214, 10:PRINT "选择处理方式";CALL 499 44
100 POKE 215, 3:POKE 214, 10:PRINT "0";"␣␣";"出口"CALL 49944
110 POKE 214, 10:PRINT "1";"␣␣";"输入": CALL 49944
120 POKE 214, 10:PRINT "2";"␣␣";"显示": CALL 49944
125 POKE 214, 10:PRINT "3";"␣␣";"修改": CALL 49944.
130 POKE 215, 8:POKE 214, 2:PRINT "选择(0～3)";
140 INPUT "␣";K
150 IF K=0 THEN 820
160 ON K GOTO 200, 300, 500
170 GOTO 80
200 READ A$, B$, C$:IF A$="␣" THEN 700
210 PRINT WR$;F$;", R";M+1:PRINT A$:PRINT B$:PRINT C$
220 M=M+1:PRINT WR$;F$;", R0":PRINT M
230 PRINT D$:GOTO 200
300 GOSUB 350
310 FOR R=R1 TO R2:GOSUB 450:NEXT R
320 PRINT D$:GOTO 700
350 INPUT "行号 0:所有行:"; Q$
360 PRINT Q$:Q=VAL(Q$):IF(Q<1 OR Q>M-1)
    AND Q$< >"0" THEN PRINT CHR$(7); :GOTO 350
370 R1=Q+1:R2=Q+1:IF Q=0 THEN R1=2:R 2=M
380 CALL 49941
390 R=1:I=2:GOSUB 450:PRINT D$
400 FOR J =1 TO 3:CALL 49944:POKE 214, (J-1)*10+3:PRINT"————";:
    NEXT J:PRINT ""
410 RETURN
450 PRINT RD$;F$;", R";R
460 FOR J=1 TO 3:INPUT ""; A$ (J):CALL 49944:POKE 214, (J-1)*10+3:
PRINT A$(J); :NEXT J:PRINT ""
470 CALL 49944:RETURN
500 GOSUB 350
510 I=2:FOR R=R1 TO R2
520 PRINT RD$;F$;", R";R:FOR J=1 TO 3:INPUT A $(J):NEXT J: PRINT
D$
525 POKE 215, I: POKE 214, 0 :PRINT R-1
530 FOR J=1 TO 3:POKE 215, I:POKE 214, (J-1)*10+3:CALL 49956:
PRINT A$(J); :POKE 214, (J-1)*10+2:INPUT Q$:CALL49944
540 IF LEN(Q$)<>0 THEN A$(J)=Q$
```

```
550 NEXT J
560 PRINT WR$;F$;", R" R:FOR J=1 TO 3:PRINT A $(J):NEXT J
570 PRINT D$: NEXT R
700 POKE 215, 9:PRINT "按空格键返回";
710 GET G$:IF G$=" "THEN 80
720 GOTO 700
750 DATA "姓名", "学号", "成绩"
760 DATA "王林", "123512", "85"
770 DATA "陈红", "123514", "90"
780 DATA "李明", "123516", "75"
790 DATA "张玲", "123518", "82"
800 DATA "林青", "123520", "92"
810 DATA " ", " ", " "
820 END
```

程序中语句 80～130 显示功能选择表，如图 4-5-4。

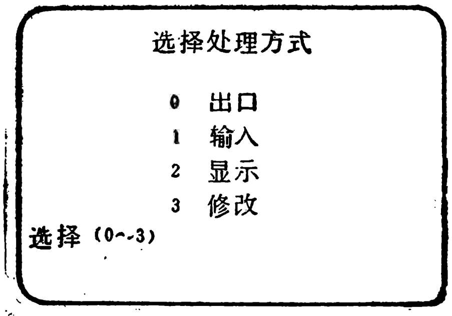


图 4-5-4

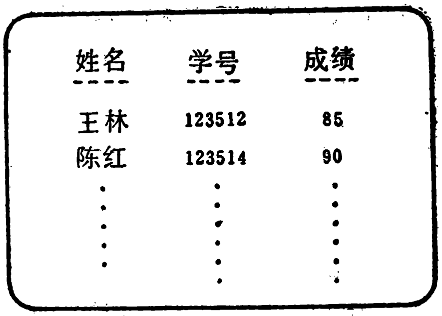


图 4-5-5

选择回答 1，读入成绩表，表格的一行作为一个记录写入文件，并在首记录中写入当前文件中最后一个记录号。

选择回答 2，转入显示成绩的处理程序，先执行语句 350～410 的子程序，在选择回答行号后（如回答"0"，显示整个表格）显示栏标题，并在栏标题下显示点划线。然后执行语句 310，显示成绩表某一行或整个成绩表。如图 4-5-5 所示。语句 450～470 子程序显示一个记录的内容。

选择回答 3 转入修改表格项目的处理程序。先显示标题及点划线，然后读出所选择的记录，在屏幕最左方显示行号，并逐项显示在相应的栏标题下面。显示某一项后光标指到这一项的开头，用→键把光标移到需修改的字符处，修改后用→键把光标移到这一项的最后，按 RETURN 键，光标指向下一项，继续修改。一个记录修改后，写入修改过的内容。

上述各处理程序最后都转到语句 700 去执行，按空格键后，返回显示功能选择。

选择回答 0，结束程序运行。

第六节　文件及软盘的复制

汉字信息处理系统程序及数据通常都保存在软盘上，为了防止意外事故而丢失这些信息，如电源断电、意外的重写、删除或者盘损坏等。有用的软盘最好保存有一个以上的副本，并在调试程序及输入数据过程中定期地复制副本，将各副本按先后编好次序，一旦发生意外，仍可使用最新拷贝的副本，使损失达到最小。

APPLE Ⅱ 提供多种方法复制各种类型的文件或整个软盘，通常称包含被复制内容的软盘为源盘，保存副本的软盘称为复制盘，在复制之前，最好对源盘进行写保护，即在盘的保护缺口上粘上胶纸，以免意外破坏源盘上的信息。

一、复制 APPLE SOFT 程序文件及二进制文件

这两种文件复制方法相似，操作过程如下：

(1) 把源盘放进 1 号驱动器，输入 LOAD 或 BLOAD 命令，将程序文件或二进制文件装入到内存。

(2) 若使用一个驱动器，复制到同一个盘片上，直接输入 SAVE 或 BSAVE 命令，但应指定与原文件不同的文件名，若复制到另一盘上，取出源盘，换上复制盘，再输入SAVE 或 BSAVE 命令完成复制操作，若要用两个驱动器进行复制，需把复制盘放进 2 号驱动器，并在 SAVE 或 BSAVE 命令中指定 D2 参数。例如：SAVE HZ，D2。

一个中文 APPLE SOFT 程序文件可以转换成二进制文件，操作过程如下：

(1) 输入 LOAD 命令把程序装入到内存；

(2) 输入 BSAVE 命令，指定程序的起始地址 $800，并给出程序长度参数，程序以二进制形式写入软盘。

调用这个文件及执行过程如下：

先用 BLOAD 命令装入这个文件，同样指定 $800 为程序存放的起始地址，执行这个程序必需使用 RUN 命令，而不能使用 BRUN 命令，关于 BSAVE、BLOAD、BRUN 命令请参阅本章第九节。

二、复制正文文件

汉字信息处理系统的特点是数据量大，数据在整个系统中占有主要的地位，丢失数据会给整个系统带来严重的后果，保存副本，并在输入过程中随时复制副本是十分重要的工作，数据是以正文文件的方式保存在盘上，在目录中以 T 表示，这种文件不能用上面所述的方法复制。

在 DOS 3.3 主盘中提供一个 FID 程序，它不仅能对所有类型的 DOS 文件进行复制，而且还具有对文件进行编目、删除、封锁、解除封锁等功能，并提供了使用一个驱动器复制软盘的方法。

要使用 FID 程序，先要引导 DOS，复制盘必须经过初始化，其步骤如下：

(1) 将系统主盘中 FID 程序调到主存并执行

BRUN　FID

屏幕上显示如下的 FID选择项：

CHOOSE ONE OF THE FOLLOWING OPTIONS
(1) COPY FILES
(2) CATALOG
(3) SPACE ON DISK
(4) UNLOCK FILES
(5) LOCK FILES
(6) DELETE FILES
(7) RESET SLOT & DRIVE
(8) VERIFY FILES
(9) QUIT
WHICH WOULD YOU LIKE?

(2) 选择回答后转入相应的处理过程

FID 程序开始执行时，没有设定使用哪个磁盘驱动器，输入的第一个命令如果不是 COPY 命令，FID 程序要求操作人员设定驱动器使用的扩充插座号码及驱动器号码，以后输入的命令就作用于这个驱动器上，直到用(7)的命令取消这个约定。使用下一个命令时，必须重新设定插座及驱动器号码，若进入 FID 程序后第一个输入的是 COPY 命令，FID 要求输入两个磁盘的插座号码及驱动器号码。

下面分述各种不同选择的处理过程。

CATALOG 查阅目录

选择回答 2，由于是进入 FID 程序后第一个命令，程序要求操作人员输入扩充插座号码及驱动器号码，然后输出目录表，这个命令的功能和 DOS 命令 CATALOG 相同。在提示信息“PRESS ANY KEY TO CONTINUE”出现后，输入任何一个键，除了 RETURN, ESC, CTRL, SHIFT 以及 RESET 以外，都会返回选择处理方式字幕。操作过程如下：

WHICH WOULD YOU LIKE? 2 CATALOG
SOURCE SLOT? 6
DRIVE?1
DISK VOLUME 254
A 006 HELLO
T 005 HUN1
T 013 HUN3
T 009 HUN4
A 015 ZYI3
PRESS ANY KEY TO CONTINUE

SPACE ON DISK

选择回答 3，显示盘上已使用过的扇区数目及可以使用的扇区数目下面操过程作中指出了已使用的扇区数目为 112，还有 448 个扇区可供使用。

WHICH WOULD YOU LIKE? 3
SPACE ON DISK
0448 SECTORS FREE

0112 SECTORS USED
PRESS ANY KEY TO CONTINUE

UNLOCK 文件解除封锁

选择回答 4，提示输入文件名，如果输入非法的文件名，FID 会继续提示，直到输入正确的文件名，接着提示用户插入软盘，并指示用 ESC 键退出或用其它任何键开始对文件进行解锁操作，操作结束显示"DONE"。如果软盘上没有该文件，则产生出错信息"NO FILES SELECTED"。操作过程如下：

WHICH WOULD YOU LIKE?4

UNLOCK FILES

FILENAME?`HELLO
INSERT DISKS, PRESS <ESC> TO RETURN TO MAIN MENU OR ANY
OTHER KEY TO BEGIN
FILENAME:HELLO
FILE HELLO
DONE
PRESS ANY KEY TO CONTINUE

LOCK 封锁文件

若要封锁一个文件，选择回答 5，操作过程和对一个文件解除封锁相似。

WHICH WOULD YOU LIKE?5

LOCK FILES

FILENAME? HELLO
INSERT DISKS, PRESS<ESC> TO RETURN TO MAIN MENU
OR ANY OTHER KEY TO BEGIN
FILENAME:HELLO
HILE HELLO
DONE
PRESS ANY KEY TO CONTINUE

DELETE 删除

该命令删除一个或一组文件，选择回答 6。如果操作顺利完成，显示"DONE"；如果文件找不到，将显示"NO FILES SELECTED"；如果文件被封锁，则显示"FILE LOCKED"，且不执行删除操作。下面是一个被封锁文件的操作过程。

WHICH WOULD YOU LIKE?6

DELETE FILES

FILENAME?ZI1
INSERT DISKS, PRESS <ESC>TO RETURN TO MAIN MENU OR ANY
OTHER KEY TO BEGIN
FILENAME:ZI1

FILE ZI1
FILE LOCKED
PRESS ANY KEY TO CONTINUE

RESET 复位

这个命令用来修改扩充插座和驱动器号，选择回答 7。取消当前的插座及驱动器号，操作过程如下：

WHICH WOULD YOU LIKE?7
RESET SLOT & DRIVE
DONE
PRESS ANY KEY TO CONTINUE

VERIFY 检测

这个命令与 DOS 中的 VERIFY 命令的功能相同，选择回答 8。当文件检测后，如果当前的插座及驱动器号也被 RESET 命令取消，要求重新定义插座及驱动器号。文件检测后，如果不能读入一个文件，显示 I/O ERROR，命令正确执行后显示文件名及 DONE。

WHICH WOULD YOU LIKE?8
VERIFY FILES
SOURCE SLOT?6
DRIVE?1
FILENAME?HELLO
INSERT DISKS。PRESS<ESC>TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN
FILENAME:HELLO
FILE HELLO
DONE

COPY 复制

选择回答 1 可以对文件进行复制。
双驱动器的复制过程如下：

WHICH WOULD YOU LIKE?1
COPY FILES
SOURCE SLOT?6
DRIVE?1
DESTINATION SLOT?6
DRIVE?2
FILENAME?ZYI3
INSERT DISKS。PRESS <ESC> TO RETURN TO MAIN MENU OR ANY OTHER KEY TO BEGIN
FITENAME:ZYI3

FILE ZYI3

DONE

PRESS ANY KEY TO CONTINUE

通常盘驱动器控制卡插在 6 号插座上，两个驱动器号不同，在文件名的提示下，输入一个文件名，则复制这个文件，输入"="，则复制一个完整的软盘，输入文件名为"FR=ED"，程序选择所有以"FR"开头，以"ED"结尾文件名所代表的文件；输入"=N="，则选择文件名中包含N的文件。输入文件名后，系统提示把源盘及复制盘插到相应的驱动器中，并按任一键开始复制；

如果源盘中没有被复制的文件，则显示错误的信息"NO FILE SECECTED"。

如果复制正常结束，显示 DONE。

如果复制盘上已存在一个文件，它和要复制的文件同名，则显示信息"FILE ALREADY EXISTS"，同时输出下列信息，提示你如何处理：

TYPE IN NEW FILE NAME FOR THE COPY OR<RETURN> TO REPLACE PRESENT FILE OR<CTRL-C><RETURN> TO CANCEL COPY

你可以作如下三种选择：

（1）重新输入一个新的文件名；

（2）用现在要复制的文件代替旧文件，输入 RETURN，若旧文件没被封锁，新文件将替代旧文件，如果旧文件被封锁，将显示

FILE LOCKED

DO YOU WISH TO REPLACE IT ANYWAY?

接着输入 Y，复制开始，旧文件被复盖。如果输入 N，系统要你重新对这三种选择作出回答；

（3）如果不再复制，可输入 CTRL-C RETURN。

单驱动器的复制过程要求输入相同的源盘和复制盘所在的扩充插座号和驱动器号，同时系统提示你轮流把源盘及复制盘放入驱动器，将文件分段读出及写入，在两盘之间进行复制。

QUIT　退出

选择回答 9，退出 FID 程序，回到 BASIC 系统。

FID 程序可以复制包含中文信息的文件，可以在英文系统下复制，如果文件名是中文，必须先用 RENAME 命令改成任一个英文名，复制后重新用 RENAME 命令改回来。

三、复制软盘

APPLE Ⅱ 系统主盘提供一个 COPYA 程序，它能将一个磁盘上的内容全部复制到另一个磁盘上，COPY A有格式化功能，复制盘不需要预先进行格式化，复制步骤如下：

（1）输入 RUN COPYA 命令把系统主盘中的 COPYA 程序装入到内存并执行，屏幕上显示：

APPLE DISKETTE DUPLICATION PROGRAM ORIGINAL SLOT:DEFAULT =6

通常驱动器控制卡插在 6 号插座，源盘放在 1 号驱动器，复制盘放在 2 号驱动器；

（2）当 ORIGINAL SLOT 信息出现时回答 6，
DRIVE 信息出现时回答 1；

（3）当 DUPLICATION SLOT 信息出现时回答 6，
DRIVE 信息出现时回答 2；

（4）正确回答上述问题后，屏幕上将出现

——PRESS 'RETURN' KEY TO BEGIN COPY——

按下 RETURN 键，复制开始，屏幕上反复出现"READ"和"WRITE"信息，其复制过程是以一次读一部分源盘上的信息，然后写入复制盘的方式进行，反复多次、直至复制结束。

（5）复制结束屏幕上出现提示信息

DO YOU WISH TO MASK ANOTHER COPY？

若需重新复制一张，可将这次复制过的软盘从 2 号驱动器取出，将另一个复制盘放入，按Y键后开始新的复制过程。

若不再复制，按N键退出 COPYA 程序。

复制过程也可以在一个驱动器上进行，重复上述(1)、(2)步骤，在(3)中的 DRIVE 后面回答 1。当(4)中的提示信息出现后，从 1 号驱动器取出系统主盘，插入源盘，并按下 RETURN键，程序读了一部分源盘信息后通知操作者用复制盘代替源盘，并初始化源盘，然后程序提示重新插入源盘，读出源盘信息，再次提示插入复制盘，把信息复制到盘上，重复后两个工作，直到源盘被复制完。

第七节　程序的连接

APPLE Ⅱ-PLUS 有 48K 字节的 RAM 存贮器，可供程序数据使用的存贮空间仅二十多K，而且存放一个实数占 5 个内存单元，存放一个整数占 2 个单元，存放一个字符串占 4 个以上的单元，通常设计一个汉字系统处理程序都需要采用覆盖技术，把一个程序分成多个程序段，存放在磁盘上，系统提供功能选择表，用户选择后调入相应的程序段执行，也可以在一个程序段中调用另一个程序段，程序段之间的连接通常可以采用以下几种方法：

一、不需要传递变量值的程序段间的连接

若各程序段间不需要传递变量值，可以用 DOS 命令进行连接，在第一段程序的最后一行写上

PRINT CHR$(4)；"RUN PART TWO"。其中PART TWO是第二段程序的文件名。

例 1　若第三章第六节例中功能选择表提供的两个功能程序段相互独立，并以各自的文件名存放在盘上，则语句 70，80 必须作如下改动，并增加语句 90，100。

```
 70 ON C GOTO 90, 100
 80 D$ = CHR$(4)
 90 PRINT D$;"RUN 查询"          "查询"是查询程序段的文件名
100 PRINT D$;"RUN 打印"          "打印"是打印报表程序段的文件名
```

调入新程序前机器自动将内存中原有的程序及数据全部清除。各程序段中最后要写上返

回功能选择程序段的语句，即 PRINT D$；"RUN 选择"。"选择"为功能选择程序的文件名。

如果两段程序需要共用一些信息，这些信息也将丢失，使第二段程序无法正确运行，解决办法如下所述。

二、需要传递变量值的程序段间的连接

1．英文系统下的连接程序 CHAIN

APPLE Ⅱ系统主盘中提供一个机器语言程序 CHAIN，它不仅可实现两段程序的连接，而且实现中间数据的传递，但这个 CHAIN 程序不能对汉字处理程序进行连接。使用CHAIN命令须将主盘中的 CHAIN 程序复制到用户程序盘上．CHAIN 的使用方法如下：

（1）输入 BLOAD CHAIN，A12296 将系统主盘的 CHAIN 程序调入内存；

（2）输入 BSAVE CHAIN，A12296，L456 将 CHAIN 存入用户程序盘；

（3）在第一段程序调用第二段程序之前写入语句

```
1000 PRINT CHR$(4);"BLOAD CHAIN, A520";
```

（4）在第一段程序的最后一个语句调用第二段程序。设第二段程序名为"PART TWO"

```
1010 CALL 520"PART TWO"
```

值得注意的是 CALL 520 与双撇号之间是不留间隔的。

（5）运行第一个程序段，即输入命令"RUN PART ONE"就可以完成程序的连接及变量的传递。

2．中文 BASIC 程序的连接

中文数据处理系统中若两个程序段间需要相互传递信息，例如两个程序段间公用变量X及带有汉字的字符串变量 F$，必须在第一段程序的最后部分把X及F$保存到受保护的缓冲区。设第二段程序的文件名为 PART TWO，第一段程序的最后部分写上如下的语句：

```
1000 POKE 768, X
1010 L = LEN(F$):POKE 769, L
1020 FOR I = 1 TO L:POKE 770 + I, ASCMID$(F$, I, 1)):NEXT I
1030 PRINT CHR$(4);"RUN PART TWO"
```

其中汉字字符串必须以中文字母的 ASC Ⅱ码存放。

在第二个程序段的起始部分，必须先取出变量及X中文字母，并恢复成汉字字符串

```
 5 F$ = ""
10 X = PEEK(768):L = PEEK(769)
20 FOR I = 1 TO L:F$ = F$ + CHR$(PEEK(774 + I)):NEXT I
          ⋮
          ⋮
```

当运行第一个程序段后自动装入第二个程序段，并把变量X及F$传递给第二个程序段。

可以使用 LOMEM，HIMEM 语句为要传递的信息预留缓冲区，详见第五章叙述。

3．RENUMBER 程序

DOS3.3 主盘提供一个 RENUMBER 程序，这个程序有两个功能。

(A) 对一个程序重排行号；

(B) 把两个程序拼接成一个程序。

RENUMBER 程序使用的方法如下：

(1) 执行 RENUMBER 程序，即输入 RUN RENUMBER 命令，屏幕显示如下：

RENUMBER (DEFAULT VALUES)

& AFIRST 10U A, INC 10U A, S OU A, E 63999U

MERGE

&H　　PUT PROGRAM ON HOLD

&M　　NERGE TO PROGRAM ON HOLD

PRESS 'RETURN' TO CONTINUE…

其中：&FIRST n 指出重排行号后，新行号的起始位，若省略了这个参数，自动设定为 10。

INC n：指出重排行号后，各行号之间的间隔，若省略这个参数，自动设定为 10。

Sn：指出要进行重排行号的程序段最小行号，省略它，自动设定为 0。

En：指出要进行重排行号的程序段的最大行号，省略它，自动设定为 32768。

&H：保持当前内存中的程序使它不受破坏。

&M：把新装入的程序和于用 &H 保持的程序拼接。

按程序的提示，用户接着按 RETURN 键，在下一个屏幕页显示：

RENUMBER IS INSTALLED AND READY

IF YOU USE 'FP', 'HIMEM', OR 'MAXFILES'

YOU WILL HAVE TO RE-RUN RENUMBER

它指出 RENUMBER 程序已准备好。

(2) 若要对一个程序重排行号，输入 LOAD 命令，把这个程序装入到内存，接着输入 &FIRST 命令，例如你要对程序中语句 100～500 重排行号，重排后起始行号为 50，语句间间隔为 10，则输入命令：

I&FIRST 50, INC 10, S100, E500

当屏幕上出现提示符]，重排工作完成。

(3) 若要将两段程序拼接成一个程序，如程序 1 和程序 2，这两个程序可能有重复或交错，对程序 1，可以不重新排行号，但程序 2 必须重新排行号，而且程序 2 的新行号的起始值一定要大于程序 1 的最大行号，否则拼接之后，交接后出现两个同行号的语句或语句交错的情况。

拼接过程如下：

在对程序 1 重排行号后输入命令 &H，用以保护当前处于内存中的程序 1，然后用LOAD 命令装入程序 2，重排行号后输入命令 &M，拼接工作即告完成。

第八节　磁盘管理信息

DOS 对一个文件的读写过程是以一个扇区 256 个字节为基本单位进行的，就是说一个文件是被分成一个个区段存贮在磁盘上的一个个扇区中，为了记住某一个文件的各个区段存放在盘上哪一个道，哪一个扇区上，DOS 在存贮文件时，也将其所用到的磁道，扇区记录在

一个表中，这个表称为道/区表(或 T/S 表)。并建立文件索引表，记录文件名称、类型，以区段为单位表头的文件长度，以及文件的道/区表存在磁盘上的位置等，对每一个磁盘 DOS 都建立一个磁盘内容表，或称 VTOC 表，它给出了该磁盘的一些状态及盘区的分配情况，这些表格就是磁盘管理信息，对磁盘的读写操作是以查表方式进行的。

一、VTOC 表

VTOC 表是磁盘的内容表，位于 17 道第 0 区，占一个扇区，信息内容如下：

字　节	内　　容
0	未用
1	第一个目录表区段的道号
2	第一个目录表区段的区号
3	DOS 的版本号
4～5	未用
6	盘卷号(1～254，初始设定 254)
7～26	未用
27	在一个道/区表区段中所能存放的道/区对的最大数(122)
28～2F	未用
30	已分配的最后一道
31	盘的分配方向
32～33	未用
34	磁盘的总道数(35)
35	每道的区段数(13或16)
36～37	每个区段的字节数
38～3B	第 0 道的各个区段分配的标志
3C～3F	第 1 道的各个区段分配的标志
⋮	⋮
⋮	⋮
C0～C3	第 34 道的各个区段分配的标志
C4～FF	未用

VTOC 表中，磁盘上每一道各个区段的分配与否的标志占用 4 个字节，字节与区段的对应关系如下：

字　节	对 应 的 区 段
0	FEDCBA98
1	76543210
2～3	未用

字节 0 及 1 中每一位对应一个区段，如果一个区段已分配给某个文件，相应位为 0，未分配为 1。

二、索引表

索引表记录了盘上每个文件的基本情况，比如文件名、类型、长度、道/区表的位置等，

它存放在第17道上，一般从第15区段开始存放，每一个索引区段可放7个文件的索引项，如文件超过7个，超出的索引项依次存放到第14区段，直至第1区段。索引表中的第1、2字节完成区段的连接，索引表最多可存取10个不同文件的索引项。

索引表在盘上存放的位置由VTOC表中的指针(1～2字节)指出，一个磁盘索引表的各个区段格式如下：

字节	字节的内容
0	末用
1	下一个索引区段的道号(通常为17道)
2	下一个索引区段的区段号
3～A	未用
B～2D	文件1的索引项
2E～50	文件2的索引项
⋮	⋮
⋮	⋮
DD～FF	文件7的索引项

一个文件的索引项占35个字节，内容如下：

字节	字节内容
0	文件的第一个道/区表区段的道号
1	文件的第一个道/区表区段的区段号
2	文件类型
3～20	文件名
21～22	文件占用软盘区段的数目

文件类型字节中每一位所代表的文件类型如下表：

字节	CATALOG命令显示的符号	文件类型
7	*	=1 文件被封锁
		=0 文件未封锁
6～3		全为 0
2	B	=1 二进制文件
1	A	=1 APPLE SOFT BASIC文件
0	I	=1 整数BASIC文件
0～6位全为0	T	正文文件

文件类型的标志字节可以为以下几种可能的值：

文件类型	类型字节的值	
	文件未封锁	文件已封锁
正文	0	80
整型	1	81
APPLE SOFT BASIC文件	2	82
二进制	4	84

文件索引项中，字节21和22的最大计数值为255。如果一个文件的区段计数值大于255，则重新从000开始计数，最后的计数值在CATALOG命令中显示出来，在查阅一个

文件占用的区段值时要注意这一点。

三、道/区表

道/区表指出一个文件各个区段在盘上存放的位置，它以文件存贮的方式存放在软盘上，每个文件都对应一个道/区表，道/区表的第一个区段在盘上的存贮位置由这个文件的索引项中的字节 0 和字节 1 指明，以后各个道/区表的区段位置由道/区表中的字节 1、2 的指针连接，道/区表的格式如下：

字节	字节的内容
0	未用
1	下一个道/区表的道号
2	下一个道/区表的区段号
3～4	未用
5～6	该道/区表区段中的第一个区段在文件中是第几个区段
7～B	未用
C	文件第一个区段的磁道号
D	文件第一个区段的区段号
E	文件第二个区段的磁道号
F	文件第二个区段的区段号
⋮	⋮
⋮	⋮
FE	文件第 122 个区段的磁道号
FF	文件第 122 个区段的区段号

当 DOS 访问一个盘文件时，先读 17 道 0 区的 VTOC 表，从表中的第 1、2 字节查到第 1 个索引区段的位置，从这个索引区段开始依照索引区段的第 1、2 字节连接指针查找各个目录区段中的文件索引项，查到从表中找到所需文件的文件索引项，从所需文件索引项的第1、2 字节查到存放第一个道/区表的盘位置，然后从道/区表中查得各文件区段所占的盘位置，如果一个文件的长度超过了 122 个区段，需要由道/区表的第 1、2 字节查出下一个道/区表区段位置，再从中查到超过的区段存放的盘位置，以此类推，直到读出整个文件，所以只要读出 17 道 0 区的 VTOC 表，就可以了解整个盘空间的使用情况，当然也知道一个文件的存贮情况，下一节我们将介绍如何读出 17 道 0 区的 VTOC 表。

第九节　机器语言文件

机器语言文件即二进制文件，它包含了内存中一个特定范围中的二进制信息的准确映象。这些二进制信息可以是机器语言程序，二进制数据或高分辨图形区的信息，在索引中，二进制文件名前用"B"表示。处理二进制文件的 DOS 命令是 BSAVE，BLOAD，BRUN。它们的功能和处理 APPLE SOFT 程序文件的 SAVE，LOAD，RUN 命令相同。二进制机器语言的程序也可以在 BASIC 程序中使用 CALL 或 USR 功能调用。

一、有关二进制代码文件的 DOS 命令

1．BSAVE 命令

BSAVE 命令建立一个二进制文件，把内存中特定区域的二进制信息存进磁盘。

格式：BSAVE f，Aa，Lj[，Ss][，Vv][，Dd]

f：文件名；

A：指定内存中要存放的二进制信息区域的起始地址。a 的范围是 0～65535；

L：指定这个二进制信息区域的长度，以字节计数，j 的范围是 1～32767，如果一个区域的长度超过 32767，必须使用两个 BSAVE 命令。

这两个参数不能省略，L、A 参数值可用十六进制表示，但十六进制值前面要加上 $。

例如：BSAVE 文件名，A$800，L2000，S6，D1，V6

它表示把从主存 $800 开始，长度为 2000 字节的二进制信息以一个指定的文件名存放到磁盘上。

2．BLOAD 命令

BLOAD 命令将二进制文件的内容送回内存。

格式：BLOAD f[，Aa][，Ss][，Dd][，Vv]

在这个命令中A 参数可以省略，这时文件的内容将送回到这个文件原来所在的内存中(即执行 BSAVE 时指定的A值)，如果指明了A参数，那么将从地址 a 开始存放，BLOAD 命令一般不会把 BASIC 程序清除，除非二进制信息被加载到包含 BASIC 程序的特定内存区域。

例如：BLOAD 文件名，A$800

它表示把一个指定的机器语言文件装入到内存 $800 开始的区域中，

3．BRUN 命令

格式：BRUN f[，Aa][，Ss][，Dd][，Vv]

只有当二进制文件 f 是机器语言程序时才可以使用这个命令。

BRUN 命令首先执行程序的装入操作，也即重复BLOAD 操作，然后它执行一个机器语言的转移指令，转到程序的入口并执行这个程序。

二、磁盘驱动程序(RWTS 子程序)

通常，用户若要访问盘文件，只能使用DOS 操作系统，而且文件在盘上存贮的位置是由 DOS 管理的，只有读出 17 道 0 区的 VTOC 表，才能了解到一个文件在盘上的存贮情况，用户不可能任意指定在某一磁道和区段上读、写信息。

DOS 以层次结构实现对系统的操作管理以及磁盘的读写操作和磁盘文件各项管理操作，DOS 的结构分成如下三个主要层次：

外层为 DOS 的主体程序，如冷/热启动入口程序、BASIC 的接口程序及 DOS 命令的解释程序；

中间一层是文件管理程序，它负责完成磁盘文件的读写和维护管理工作；

内层磁盘驱动程序，也称“RWTS”程序，即“区段读写程序”，是 DOS 的内核程序，它完成基本的磁盘读写操作，也就是说，它以区段为单位对磁盘进行读写。

三层之间接口明确，外层可以通过一定的命令表格调用内层的程序模块，RWTS 是作

为一个子程序在 DOS 中由文件管理程序调用。

在盘上的读写操作是以一个扇区为单位的。每个扇区分成两个域：地址域和数据域。地址域包括盘卷号、道号、区号。数据域包含 256 个字节数据的实际代码(信息是以一定的编码方式记录在盘上)。当 DOS 访问盘文件时，通过查 VTOC 表、磁盘索引表及道/区表，找到文件各个区段在磁盘的哪个磁道，哪个扇区上。磁盘控制器驱动磁头定位到磁道后，从当前位置开始找下一个地址域，并读出地址信息进行比较，如果不是要查找的扇区，则再读下一个扇区的地址域，直至找到所要的扇区，再完成读或写操作。每次读写一个扇区的操作正是由 RWTS 子程序完成的。可以看到，如果用户也能直接调用这个子程序，用户就能指定在盘上的任意磁道和扇区上读，写信息。APPLE Ⅱ 提供了这一功能。

DOS 操作系统是由机器语言编写的，所以 RWTS 子程序也是一个机器语言程序，其入口地址是 $03D9，当 DOS 中的文件管理程序调用它时，已经把执行 RWTS 子程序所需的信息建立在两张表格中，它们是 I/O 控制表和设备特征表。

IOB 表，即输入/输出控制块表，它向 RWTS 子程序提供：磁盘驱动器所在的槽号及具有的驱动器号、盘的卷号，要访问的磁道和扇区，以及读或写软盘的标志，设备特征表向 RWTS 提供操作 DISK Ⅱ 所必需的信息，并将此 I/O 控制表的地址放入 Y 和 A 寄存器。

这样用户直接调用 RWTS 子程序，必须首先建立 IOB 表及设备特征表，APPLE Ⅱ 提供一个机器语言程序供用户建立这两个表。

这两个表的内容如下：

1. IOB 表格式

字节($)	内容
0	表类型，定为 $01
1	扩充插座号 *16
2	驱动器号
3	卷号
4	要进行存取的道号
5	要进行存取的扇区号
6	设备特征表的低位字节
7	设备特征表的高位字节
8	256 个字节读写缓冲区地址的低位字节
9	256 个字节读写缓冲区地址的高位字节
A～B	未用
C	命令码(0：查找，1：读区段，2：写区段，4：格式化区段)
D	指出执行期间的错误类型($10：软盘写保护，$20：卷号不匹配，$40：驱动器错，$80：读错误，允许重读 48 次)
E	上次访问时用的卷号
F	上次访问时用的扩充插座号*16
10	上次访问时用的驱动器号

2. 设备特征表格

字节($)	内容
0	设备类型，使用 DISK Ⅱ，必须为 $00

1	每道步进道数，为 $01
2~3	马达运转时间计数的补码，字节 2 中必须是 $EF，字节 4 是 $D8。

按上述要求建立表格后，用户要使用一个控制子程序，将 IOB 表的地址存放到 Y 和 A 寄存器中，A 寄存器存放高位地址字节，Y 寄存器存放低位地址字节，并转 RWTS 的入口地址 $3D9。

例：试用 RWTS 子程序读出 17 道 0 区的 VTOC 表。

第一步：建立输入/输出控制表及设备特征表，假设表格由内存 $ C0A 单元开始存放，使用的扩充插座是 6，驱动器号是 1。把 17 道 0 区的 VTOC 表读到从内存 $2000 开始的存贮区中；

IOB 表

单　元	内　容	说　明
$C0A	01	表类型
$C0B	60	扩充插座号 6*16
$C0C	01	驱动器号为 1
$C0D	00	卷号
$C0E	11	读取 17 道
$C0F	00	读取 0 扇区
$C10	20	设备特征表的低位字节
$C11	0C	设备特征表的高位字节
$C12	00	读到内存 $2000 开始的存贮区
$C13	20	
$C14	00	未用
$C15	00	未用
$C16	01	命令码， $01＝读
$C17	00	出错码
$C18	00	实际卷号
$C19	60	前一次进行存取的扩充插座号
$C1A	01	前一次进行存取的驱动器号

设备特征表紧接着 IOB 后给出

$C20	00	设备类型码
$C21	01	每道步进道数
$C22	EF	}马达运转时间计数
$C23	D8	

第二步：用机器语言编写控制子程序，它由 $C00 开始存放；

单　元	内　容	汇编语言	说　明
$C00	A90C	LDA＃$0C	IOB 表地址高位字节送A寄存器
$C02	A00A	LDY＃$0A	IOB 表地址低位字节送Y寄存器
$C04	20D903	JSR $03D9	转 RWTS 子程序
$C07	60	RTS	
$C08	00	BRK	

第三步：启动控制子程序，执行读操作。

在监控系统中可以使用 *C00G 命令启动。

在 BASIC 中可以用 CALL 3072 语句调用。

关于机器语言程序及表格存放在内存单元中的操作过程以及监控系统命令见监控系统中的叙述。

读出 17 道 0 区到主存 $2000 开始的存贮区域后，可以在监控系统下显示或打印 VTOC 表，其操作请参阅监控系统中的有关叙述。

写盘操作和读盘操作类似，通常所要修改的仅仅是 IOB 表中的命令码，内存的起始地址以及磁道号、扇区号。

第五章　监 控 系 统

APPLE Ⅱ PLUS 微机的监控系统是一个高效率的监控系统，它具有管理存贮器、设置外部设备、设定显示方式、对屏幕进行编辑、简单的算术运算及反汇编等功能。监控程序占 2K 字节的内存容量，存放在 ROM 的最高地址区域 \$F800～\$FFFF 中，这个监控系统具有自启动的特点，当初始开机通电，即所谓"冷起动"时，如果外设扩充插座上插有磁盘驱动器控制卡，监控系统会自动启动驱动器进入盘片上的 HELLO 程序，如果没有盘片，监控自动进入英文 APPLE SOFT BASIC。

这一章叙述在调试程序时常用的管理存储器的功能及反汇编功能。有关屏幕的编辑功能及可执行的控制字符已在第二章中叙述过。

第一节　存储器的组织

APPLE Ⅱ 的存贮器分成三个主要部分：随机存贮器 RAM，只读存贮器 ROM 及输入、输出(I/O)缓冲存贮器。

APPLE Ⅱ 使用 6502 微处理器，地址总线 16 位，可以访问 65536 个内存单元，这些内存单元划分为 256 页，每页 256 个单元，其中十六进制的地址前两位表示内存单元所在的页号，后两位表示在一页内的位移。

一、APPLE Ⅱ 存贮器的组织

APPLE Ⅱ 存贮器的组织如图 5-1-1 所示。

1．随机存贮器 RAM

RAM (48K)
I/O (2K)
I/O ROM (2K)
ROM (12K)

图 5-1-1

存贮器地址		用途	
十进制	十六进制		
0～255	\$0000～\$00FF	系统工作单元	
256～511	\$0100～\$01FF	堆栈区	
512～767	\$0200～\$02FF	键盘读入缓冲区	
768～1023	\$0300～\$03FF	监控程序的转移地址	
1024～2047	\$0400～\$07FF	第一文本区或低分辨率图形区	
2048～3071	\$0800～\$0BFF	第二文本区或第二低分辨率图形区	可使用的RAM
3072～8191	\$0C00～\$1FFF		
8192～16383	\$2000～\$3FFF	第一高分辨率图形区	
16384～24575	\$4000～\$5FFF	第二高分辨率图形区	
24576～49151	\$6000～\$BFFF		

图 5-1-2

RAM 共 48KB，附加 16KB 的 RAM 扩充卡可扩充到 64KB。RAM 的组织及使用情况如图 5-1-2 所示。

2．只读存贮器 ROM

主机上只读存贮器驻留 12KB，地址分配为最高的地址区域 $D000～$FFFF，在 ROM 中主要包括具有自动启动功能的监控程序及 APPLE SOFT BASIC 解释程序，如图 5-1-3 所示。

存贮器地址		用　　途
十进制	十六进制	
53248～63487	$D000～$F7FF	APPLE SOFT BASIC 解释程序
63488～65535	$F800～$FFFF	自动启动监控程序

图 5-1-3

3．I/O 存贮器

由于 6502CPU 没有专用的 I/O 指令，故采用访内指令访问外设，I/O 存贮器地址是 $C000～$CFFF，分配如下：

$C000～$C7FF 用于主机板及接口卡外设；

$C800～$CFFF 用于各接口卡的 ROM 扩充。

APPLE Ⅱ 主机电路板上有 8 个扩充插座，除 0 号插座外，其余 7 个都可接外部设备卡。每个扩充插座都分配有 16 个存贮单元作为一般输入、输出用，其地址为 $C080～$C0FF。每个外设的输入、输出存贮区如图 5-1-4 所示。

外设卡输入输出存贮区	插座编号
$C080～$C08F	0
$C090～$C09F	1
$C0A0～$C0AF	2
$C0B0～$C0BF	3
$C0C0～$C0CF	4
$C0D0～$C0DF	5
$C0E0～$C0EF	6
$C0F0～$C0FF	7

图 5-1-4

对应每个插座有一页 ROM 存贮区（256字节）存放这个外部设备的驱动程序或子程序，这一页 ROM 的地址是 $Cn，其中 n 为扩充插座的编号。插座 0 没有一页 ROM 和它对应，所以不能用来插汉卡及其它外部设备卡。外设控制卡相应的 ROM 地址见图 5-1-5。

此外，另有 2K 的 ROM 存贮区作为所有外设控制卡的共用区。其地址从 $C800到 $CFFF。尽管这个空间每个外设控制卡都可使用，但一次只能为一个外设卡使用，作为这个外设

控制卡驱动程序的扩充使用空间。

外设控制卡的 ROM 区域	插座编号
$C100～$C1FF	1
$C200～$C2FF	2
$C300～$C3FF	3
$C400～$C4FF	4
$C500～$C5FF	5
$C600～$C6FF	6
$C700～$C7FF	7

图 5-1-5

二、使用 C-PLUS Ⅱ-A 汉卡时存贮器的组织

使用 C-PLUS Ⅱ-A 汉卡时存贮器的组织如图 5-1-6 所示。其中汉卡占用的内存区域有两个。

1. $4000～$5FFF

相当于英文系统中高分辨率图形第二页的显示区，用于中文字幕显示。

2. $9000～$95FF汉卡组字区域

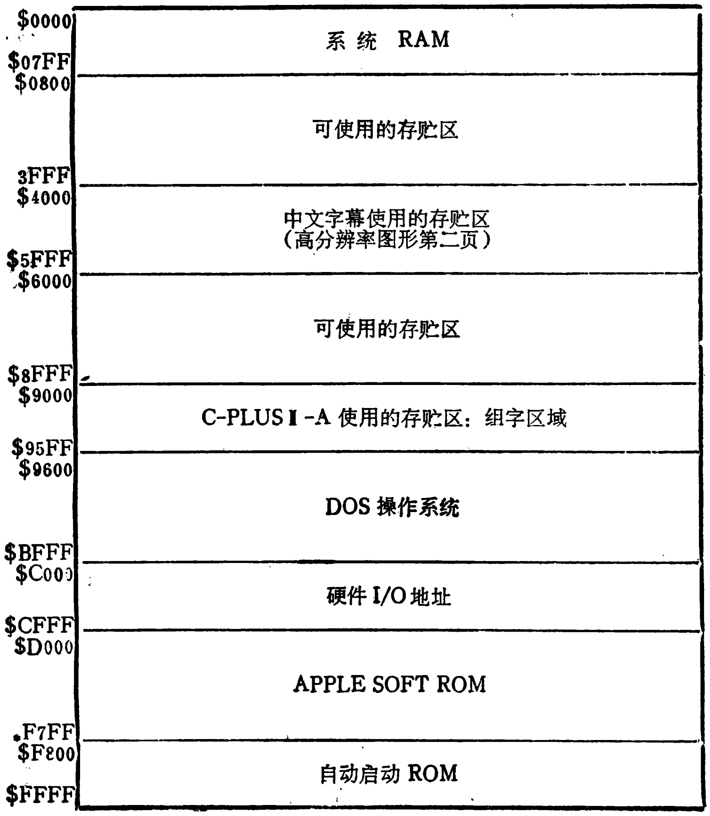


图 5-1-6

DOS 操作系统装入到主存的 $9600～$BFFF 区域。

$0800～$3FFF 及 $6000～$8FFF 这两个存贮区域是用户可使用的区域。

汉卡使用的 I/O 存贮区视汉卡插在哪个扩充插座上而定。当汉卡插在 3 号扩充插座上，汉卡驱动程序不仅使用了自己的一页 ROM 序贮区 $C300～$C3FF，而且使用了 $C800～$CFFF 这个外设控制卡共用的 2K 字节ROM 存贮区。用下面将要叙述的反汇编程序可以打出汉卡驱动程序的汇编清单。

第二节 APPLE SOFT 程序使用的存贮器

这一节介绍 APPLE SOFT BASIC 程序使用存贮器的情况。若 RAM 为 48K，DOS 操作系统常驻 $9600～$BFFF 内存区域。

一、定义变量缓冲区的 APPLE SOFT 语句

APPLE SOFT 提供了重新分配变量缓冲区的语句 HIMEM 和 LOMEM。

1. LOMEM

定义变量缓冲区的下限地址。

格式：LOMEM：地址　　其中地址须用十进制表示，其值必须小于 65535。

监控系统及 BASIC 解释程序把比 LOMEM 小的存贮区留给打印机、低分辨率图形以及屏幕显示等使用，当 APPLE SOFT 程序从磁盘加载到存贮器时，它存放在比 LOMEM 地址小的存贮区域。输入一个程序或修改程序时，LOMEM 值自动随着调整。

可以使用 LOMEM 语句定义变量缓冲区的下限地址，多留出一些存贮单元，以便给机器语言子程序或高分辨率的图形表使用。

用 NEW 命令或 CTRL-B 清除一个程序，LOMEM 值也随着改变。

在主存的 0 页 $69～$6A 单元保存了 LOMEM 值。可以在监控系统中输出 $69 及$6A 单元内容，显示当前 LOMEM 值，也可以用 PRINT PEEK(106)*256＋PEEK(105) 命令显示 LOMEM 值。

2. HIMEM

定义变量缓冲区的上限地址。

格式：HIMEM：地址

当 DOS 加载之后，机器自动设定 HIMEM 值，把 DOS 置于 HIMEM 地址以上的空间，对具有 48K RAM 的机器，在英文系统中，HIMEM 是 $9600，使用汉卡时，汉卡组字区域占用了一部分存贮空间，HIMEN 下移到 $8FFF。如果使用 DOS 中的 MAX FILES 为文本文件留出缓冲区，每一个额外的缓冲区会使 HIMEM 向下移 595 个字节。用户也可以自己设定 HIMEM 值，在 HIMEM 之上留出额外的空间来保存有用的机器语言子程序、高分辨率的图形表或两段程序共用的数据。

NEW，RUN 及 CLEAR 均不会改变 HIMEM 的值。

在主存的 0 页 $73～$74 单元保存了 HIMEM 值。可以在监控系统下显示 HIMEM 值，也可以使用 PRINT PEEK(116)*256＋PEEK(115)命令显示 HIMEM 值。

如果把 LOMEM 值定义得高于 HIMEM 值，或低于现有的 LOMEM 值，将导致存贮器不够用的错误信息：

?OUT OF MEMORY ERROR

二、APPLE SOFT 程序使用的存贮器

APPLE SOFT 程序由地址 $800 开始往高地址方向存放，并使 LOMEM 值自动往上调整。

简单变量及字符串变量的指针由 LOMEM 所指的地址开始往上存放，即置于程序之上。

数组及字符串数组的指针置于简单变量及字符串变量指针之上，字符串的值由 HIMEM 所指的地址开始往下存放，并自动调整 HIMEM 值。

APPLE SOFT 使用存贮器情况如图 5-2-1 所示。

在 0 页保存有存放这些指针的起始单元地址及结束单元地址。

$67～$68：指示程序在内存中存放的起始地址。地址高位在 $68 单元，低位在 $67 单元；

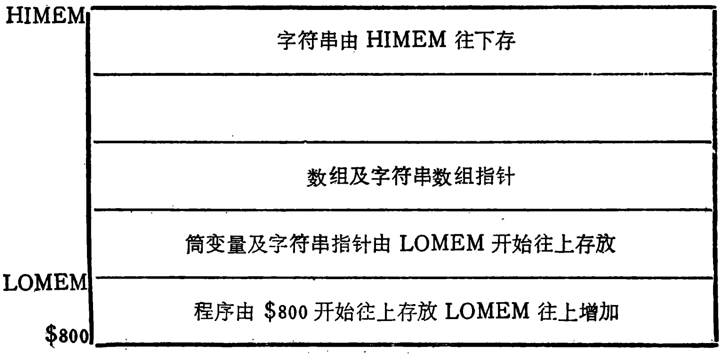


图 5-2-1

$69～$6A：存放 LOMEM 值，即存放变量及字符串指针的起始地址；

$6B～$6C：指示存放数组及字符串数组指针的起始单元地址；

$6D～$6E：指示存放数组及字符串数组指针的结束单元地址；

$6F～$70：指示当前字符串存放的起始地址；

$73～$74：存放 HIMEM 值；

在监控系统中输出这些单元中的内容可以了解一个程序运行中使用存贮器的情况。

三、中文 APPLE SOFT BASIC 程序使用的存贮器

中文 APPLE SOFT 程序使用存贮器的情况如图 5-2-2 所示。

由于汉卡组字区域使用 $9000～$95FF 区域，使 HIMEM 往下调整为 $8FFF。

中文字幕显示缓冲区使用 $4000～$5FFF 区域，使用户可使用的 RAM 分割成二部分：

$0800～$3FFF 区域存放中文 BASIC 程序及变量、字符串指针、数组及字符串数组指针；

$6000～$8FFF 区域存放字符串值。

这样，若程序大于 14K，它将侵入中文字幕显示区域 $4000～$5FFF，造成程序运行错误或出现混乱的屏幕显示图形。如果变量及数组定义过多过大，即使程序小于 14K，仍有可能侵入中文字幕显示存贮区。

若数据量很大，变量及字符串空间相对不足，也会侵入中文字幕显示缓冲区。

所以，如何合理使用存贮器，分配程序及变量空间是设计汉字信息处理系统要考虑的一个重要问题，可以使用 HIMEM 及 LOMEM 语句重新分配变量空间，比如把 LOMEM 设定为 $6000，如图 5-2-3 所示。

若系统程序很大，通常采用覆盖技术，把暂时没有用到的信息存放在盘上，需要时调入

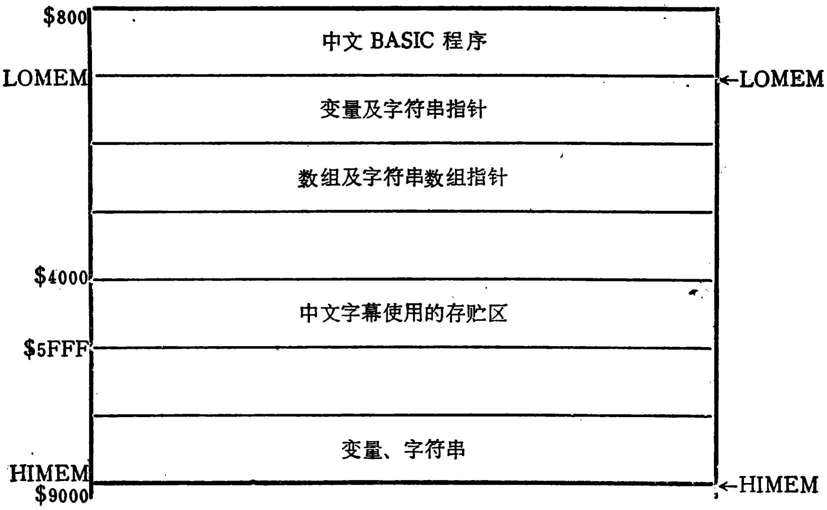


图 5-2-2

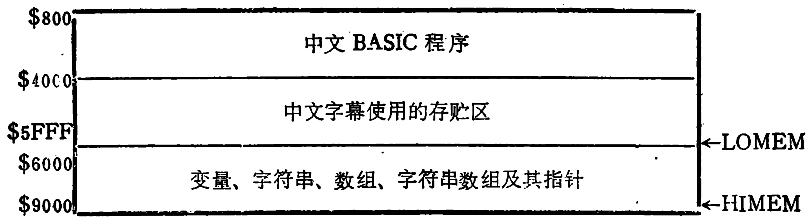


图 5-2-3

内存。关于多段程序间的链接可参阅第四章。

第三节 监控命令

这里仅给出管理存贮器及 CPU 中五个寄存器的命令，存贮器的地址以十六进制表示。

1．显示某一地址单元中的内容

命令格式：*地址 RETURN

例 1 显示 69H 单元中的内容使用如下的命令：

```
]CALL-151
*69
0069-47
*
```

显示内容“47”后，紧接着在下一行显示*。

2．显示由当前地址到满模 8 的 8 个单元中的内容

命令格式：*RETURN

例 1 中显示 69H 单元中内容之后，监控系统记忆这个地址，作为以后检查存贮器的起始位

置。在第一个*后按 RETURN 键，继续显示由当前地址到满模 8 的前一地址单元中的内容。在第二个*后，每次按 RETURN 键，顺序显示 8 个单元的内容。

例 2　显示从例 1 的当前地址到满模 8 的单元中的内容，操作如下：

```
*
 2A 55 2A 55 2A FF
*
 0070-8F FF 8F 00 90 FE FF FE
*
 0078-01 E5 0B FF 00 00 08 04
*
```

由 $67～$74 单元中的内容，可以了解到用户程序使用存贮器的情况。这里是一个用户程序装入之后而尚未执行时所显示的 $67～$74 单元中的内容。

$67～$68 是程序在内存中存放的起始地址，为 $801，$69～$6A 存放 LOMEM 值，$69 单元存放 LOMEM 的低位字节，$6A 存放高位字节，它为 $2A47，可见用户程序占用内存 $801～$2A47 的区域，由于程序尚未运行，所以 $6D～$6E 与 $6B～$6C 中均为 2A 55。$73～$74 指出 HIMEM 为 $9000，而 $6F～$70 的 8FFF 指出字符串存放的起始位置。

3．显示一个存贮器区域中的内容

命令格式：*地址 1 地址 2　RETURN

例 3　试查阅汉字字符串存贮区中 8FAOH～8FDOH 的内容。操作实例如下：

```
*8FA0.8FD0
8FA0— 68 64 62 68 67 20 69 74
8FA8— 65 20 6D 6D 6F 6B 20 65
8FB0— 69 72 20 6D 68 77 6C 20
8FB8— 62 75 20 75 75 20 68 71
8FC0— 6D 20 6F 71 20 62 20 76
8FC8— 68 71 6D 20 6E 69 72 20
8FD0— 79
```

4．显示从当前地址到某一地址区域的内容

命令格式：*·地址　RETURN

例 4　试查阅从例 3 中的当前地址开始到 8FF0H 区域中的内容。操作实例如下：

```
*·8FF0
8FD1— 72 6E 6C 20 61 6E 20
8FD8— 62 6D 72 20 65 79 78 20
8FE0— 6B 20 68 62 6E 64 20 76
8FE8— 73 6D 62 20 76 79 72 6E
8FF0— 6C
```

从地址 8FD8H 到 8FE6H 的内存区域存放了“同济大学”编码字母的十六进制 ASCⅡ值。

5．修改存贮器某一单元的内容

命令格式：*:新内容　RETURN

例 5　试将 6000H 单元中的内容修改为“同”字的第一个编码字母的 ASCⅡ码“62”，本

例的操作中，第一个*6000命令显示当前6000H单元中的内容，*:62命令把6000H单元内容修改为62，第二个*6000命令检查修改后的6000H单元内容。

```
*6000
6000-79
*:62
*6000
6000-62
*
```

6. 修改某一个存贮区域中的内容

命令格式：*:新内容1␣新内容2␣……RETURN

例6 试把7000H～7003H中的内容修改为"同"字的编码中文字母ASCⅡ码"62 6D 72 20"。修改操作如下：

```
*7000
7000-FF
*:62 6D 72 20
*7000
7000-62
*
6D 72 20 FF FF 00 00
*
```

7. 复制一个存储区域中的内容

命令格式：*地址1<地址2.地址3M RETURN

执行这个命令把地址2到地址3这个存贮区域中的内容复制到以地址1为起始地址的存贮区中。

地址2必须小于或等于地址3，如果地址2大于地址3，监控系统只把由地址2算起的一个字节移到地址1，然后结束这条命令的执行。

例7 试清除$401～$501。

先将0存到这个存贮区域的第一个地址单元中

```
*401
0401-FF
*:00
```

然后执行下面这个命令：

```
*402<401·500M
```

监督系统先将$401中的内容，即00移到$402，当移动第二个字节时，监督系统取出$401中的内容(这时已为00)送到$402，这个过程重复执行直到$500中的内容移到$501时为止。使用命令*401·501，可以检查上述命令执行结果。

如果要用一个特定的ASCⅡ码填满这个存贮区域，只要把这个值预先存到这个存贮区域的第一单元中，然后执行相同的过程。

例8 试用ASCⅡ码62，6D，72，20填入$400～$4FF的内存区域中。

首先把这四个值输入由$400起始的4个字节中：

*400

0400－FF

*:62 6D 72 20

再输入命令：

*404＜400·4FBM

原始存贮区的结束位置等于将被填满的存贮区最后一个单元地址减去上面设定的内容长度之差。

例 9　试把 $8FA0～$8FF0 的内容复制到 $400 开始的存贮区中。

输入命令：*400＜8FA0·8FF0M　执行复制过程。

8．比较存贮器中的内容

命令格式：*地址 1＜地址 2·地址 3V RETURN

执行这个命令，把地址 2 到地址 3 中的内容和地址 1 开始的存贮区内容依次逐个比较，直到所有单元中的内容比较完为止，如果正确，在比较命令下面一行显示一个*，如果不正确，则显示其出错地址以及两个不相同的值。出错内容显示在前，参考内容紧接着在括号中显示出来。

例 10　试比较 $7000～7003 和 $8FD8～$8FDB 两个存贮区中的内容。

从例 4 及例 6 知道这两个存贮区中内容都是 62 6D 72 20

*7000·7003

7000-62 6D 72 20

*8FD8·8FDB

8FD8-62 6D 72 20

输入比较命令：

*8FD8＜7000·7003V

结果正确

*

现将 8FD9 修改为 6B

*8FD9

8FD9-6D

*:6B

输入比较命令：

*8FD8＜7000·7003V

比较结果不正确，显示错误地址及内容

7002-6D(6B)

当把内存中的程序存到盘上后，可以把这个程序重新读到内存中另一个存贮区，比较这两个存贮区内容，就可以判别盘读、写是否出错，这时，比较命令是很有用的。

例 11　试将 $800 开始的汉字处理程序存入磁盘，重新读出到 $6000 开始的存贮区后，比较两个存贮区的内容。

从汉字系统使用存贮区的图示中，已知道 $6000 开始的存贮区是分配给变量及字符串变量，当程序还没执行时，这个存贮区是可以使用的，所以把盘上的内容重新读到这一区域。比较过程如下：

第一步：在 DOS 支持下输入 BSAVE 命令，把程序存到盘上

BSAVE 文件名，A$800，L$1800

第二步：输入 BLOAD 命令，把程序读回内存

BLOAD 文件名，A $6000

第三步：进入监控系统

CALL-151

*

最后，输入比较命令：

*800<6000、8000V

如果没有不相符的内容出现，可以确定程序已被正确写入磁盘了。

9. 检查和改变 CPU 中寄存器的内容

监督系统在内存中保留 5 个位置($45～$49)，用以存放 CPU 中的寄存器 A，X，Y，P 和 S 的内容，其中：

A：累加器，保存八位算术和逻辑运算结果；

X，Y：两个变址寄存器，存放变址寻址方式中的地址位移量；

P：状态寄存器，保存 CPU 的状态，状态字有

NV•BDIZC。

其中 C—进位标志；

Z—零标志，执行结果为 0，Z = 1；

I—中断禁止标志，I = 1 禁止中断；

D—十进制运算标志，D = 1 为十进制运算；

B—断点中断标志；

V—溢出标志，V = 1，结果溢出；

N—符号标志，N = 1 为负。

在监督系统中使用 CTRL—E 可以检查寄存器

*CTRL-E RETURN

显示

A = CD X = B1 Y = D8 P = B5 S = F0

监控系统把含有寄存器A 内容的存贮地址作为可修改的入口地址，所以，如果要改变寄存器 Y 中的内容，必须依次打入A 及X 的正确内容，再打入 Y 的修改值：

*:CD B1 C3

再输入 CTRL—E 命令：

*CTRL-E RETURN

监控系统修改寄存器 Y 中的内容，显示

A = CD X = B1 Y = C3 P = B5 S = F0

第四节 反 汇 编

机器语言是计算机上效率最高的语言，APPLE Ⅱ PLUS 的 DOS3.3 系统程序是用机器语言编写的。固化的系统监控程序以及外设的驱动程序也都是用机器语言编写的。当一个用

机器语言编写的程序存放在内存时，可以使用第三节中的监控命令把内存中的程序显示或打印出来，但是，由于指令都是用16进制形式表示，且各个指令包含的字节数不相同，需要逐条查对才能理解它。所以象汉卡驱动程序这样较大型的程序，要从机器语言程序中了解它是十分困难的。汇编语言程序比机器语言程序容易读，容易理解，也容易编制。

在 APPLE 监控系统中提供反汇编子程序，它可以把机器语言转变为汇编语言的格式显示或打印。

命令格式：

(1) *地址 1L RETURN

这个命令将地址1开始的20条机器语言指令转变为汇编语言格式显示在屏幕上，如果打印机已接通，打印输出，然后返回监控。

例1 打印汉卡驱动程序的汇编语言程序。

```
POKE 1403, 1
]CALL-151
*C300L
C300-   18              CLC
C301-   90 30           BCC   $C333
C303-   38              SEC
C304-   80 20           BCS   $C333
C306-   4C 91  C8       JMP   $C891
C309-   4C 9A  C8       JMP   $C89A
C30C-   4C A3  C8       JMP   $C8A3
C30F-   4C AC  C8       JMP   $C8AC
C312-   4C B5  C8       JMP   $C8B5
C315-   4C C9  C8       JMP   $C8C9
C31B-   4C CF  C8       JMP   $C8CF
C31B-   4C D5  C8       JMP   $C8D5
C31E-   4C DB  C8       JMP   $C8DB
C321-   4C E1  C8       JMP   $C8E1
C324-   4C E7  C8       JMP   $C8E7
C327-   4C ED  C8       JMP   $C8ED
C32A-   4C F3  C8       JMP   $C8F3
C32D-   4C F9  C8       JMP   $C8F9
C330-   4C BE  C8       JMP   $C8BE
C333-   48              PHA
```

值得注意的是，这里输出的是20条机器指令，由于机器指令可能是单字节、双字节或三字节的，所以在监控系统中把命令中的地址1送到 PC 计数器(程序计数器)，再由字节长度调整 PC 值，并进行计数，直至输出20条指令后返回监控系统。

(2) *L RETURN

在*地址 1L 命令执行之后，PC 计数器指向下一个指令的存贮器地址，所以尽管这个命令没有设定显示的起始地址，监控系统仍会将接下去的20条指令显示或打印出来。

```
•L
C334-   8A          TXA
C335-   48          PHA
C336-   2C FF CF    BIT   $CFFF
C339-   78          SEI
C33A-   20 8A C8    JSR   $C88A
C33D-   BA          TSX
C33E-   BD 00 01    LDA   $0100, X
C341-   58          CLI
C342-   8D F8 07    STA   $07F8
C345-   29 0F       AND   #$0F
C347-   AA          TAX
C348-   B0 25       BCS   $C36F
C34A-   8D 69 94    STA   $9469
C34D-   A9 00       LDA   #$00
C34F-   9D 78 05    STA   $0578, X
C352-   8A          TXA
C353-   0A          ASL
C354-   0A          ASL
C355-   0A          ASL
C356-   0A          ASL
```

第六章　超级汉卡

S.C.C.G-83 超级汉卡是 C-PLUS Ⅱ-A 汉卡的改进型，支持它的操作系统是 CCDOS，存放在专用的超级汉卡系统盘上。为叙述方便，我们把 C-PLUS Ⅱ-A 汉卡称为现有汉卡。

超级汉卡系统提高了屏幕显示的利用率，简化了汉字输入方法，对中文码作等长处理，解决了现有汉卡中由于中文编码长度不等导致的编程不便等问题。扩充了现有汉卡的功能，如增加了 22 种字母组输出供用户选用，其中包括英文、日文、德文、俄文、美术字等，可使打印的报表更美观，扩充了和英文 BASIC 兼容的语句 SPC，TEXT，POS，TAB，HTAB，VTAB，HOME，INVERSE，NORMAL等，并加强了现有汉卡中的PRINT功能，使它和英文系统中的 PRINT 功能兼容。在超级汉卡系统中不能使用的英文 BASIC 语句有 7 个，它们是：

FLASH，GR，PLOT，HLIN，VLIN，COLOR，SCRN。

其中除了 FLASH 外，全是绘图语句。

在超级汉卡系统中还提供了中文 BASIC 与英文 BASIC 系统相互转换的语句，它们可以直接用在程序中。

超级汉卡字母组常驻内存，若使用的系统盘未经重新选择字母组，字母组表格占用 $7200～$9000 存贮空间，从而使用户可使用的变量、数组存贮区较之现有汉卡减少了很多。为此，系统允许用户选择高分辨率第一页(PG1)存贮区($2000～$3FFF)或高分辨率第二页(PG2)存贮区($4000～$5FFF)作为中文字幕存贮区，以便调整程序存贮区与变量、数组存贮区的比例。

第一节　超级汉卡系统盘

超级汉卡系统盘上的文件索引如下：

```
]CATALOG
C1982 DSR A#254
*A  106  CCDOS
 B  003  IDSTAMP
*A  003  PRINT ALL SET
*B  002  RETCCDOS
*B  005  SETA
*B  005  SETB
*B  005  SETC
*B  005  SETD
*B  005  SETE
*B  005  SETF
*B  005  SETG
```

*B　005　SETH
*B　005　SETI
*B　005　SETJ
*B　005　SETK
*B　005　SETL
*B　005　SETM
*B　005　SETN
*B　005　SETO
*B　005　SETP
*B　005　SETQ
*B　005　SETR
*B　005　SETS
*B　005　SETT
*B　005　SETU
*B　005　SETV
*A　016　DEMO
*A　006　MODIFY
*A　005　NEWDEMO

1．CCDOS

CCDOS 是支持超级汉卡的操作系统，开机启动时，它被装入执行，直接进入超级汉卡系统状态。在索引表上，它是以符号 A 为标志的 APPLE SOFT 程序。在提示符]下，可以输入 RUN CCDOS 命令装入执行。

2．DEMO

它是一个程序文件，包含超级汉卡的使用说明，用 RUN DEMO 命令调用。程序中，每个屏幕页输出显示语句之间插入一个延时循环语句，使得每个屏幕页显示一定时间，以利阅读。整个说明反复显示，用户不再阅读时，可以用 CTRL-C 中断程序运行。如果要打印这个使用说明，输入如下语句：

```
] LOAD DEMO
] POKE 1659, 2
] POKE 2043, 100
] POKE 1403, 1
] RUN
] POKE 1403, 255
```

3．NEWDEMO

这是一个程序文件，对 DEMO 的说明作了如下的补充：

(1) 补充说明日文、俄文、美术字的字体；

(2) 超级汉卡系统与现有汉卡相互转换的说明；

(3) 在同一个程序中可以自动完成中文系统与英文系统之间相互转换的说明。

4．SET A～SET U

SET A～SET U 是超级汉卡系统中增加的 22 种字母组的表格文件，它们都是不可执行

的二进制文件。

5．PRINT ALL SET

这是一个程序文件，执行这个程序，先打开打印机，选取打印格式及打印字体等参数，然后逐一装入 SET A～SET U 表格文件，显示并打印各个文件表目及相应的字母组。打印结束，关闭打印机。显示及打印的字体依赖于当时用户所选择的 22 种字母组中的第 0 组及第 1 组，表目 SET A～SET U 以第 0 组字体打印，22 种字母组表格以第 1 组字体打印。（详细见下面叙述）。

6．IDSTAMP

这是一个不可执行的二进制文件，它记录了用户选用的中文字幕显示页(PG 1 或 PG 2)及字母组(SET A～SET U)。系统允许用户一次最多选取 10 组、各组编号依选择输入的顺序分别为 0～9，其中第 0 组定义为开机预定的文字、或重新装入 CCDOS 时屏幕上显示及打印的字符格式。这个文件在下述的 MODIFY 程序运行时以 BLOAD 命令调用，用户可以任意修改这个文件，选择合适的中文字幕显示页及字母组，修改结束后以 BSAVE 命令重新写入。在 CCDOS 装入时，根据文件中记录的用户一次选用的字母组数，CCDOS 调入相应的字母组表格。

7．MODIFY

这是一个修改 IDSTAMP 文件的 APPLE SOFT 程序文件。程序执行时，先装入 ID.STAMP 文件，然后依次提示用户选择中文字幕显示的缓冲区，选择第 0 组及开机预定的字母组以及选择增加几组字母组等，并把这些参数重新写入 IDSTAMP 文件，同时存贮到以 896 起始的存贮区，以便 CCDOS 调用，最后执行 CCDOS 程序。

8．RETCCDOS

这是一个二进制文件。当现有汉卡系统进入超级汉卡系统，须先装入这个文件。

第二节　系统修改

超级汉卡操作系统具有自我修改的能力，可适合多种不同用途。

系统修改包括下面三部分：

1．汉字字幕显示缓冲区的选用。使用者可选用 PG1 或 PG2；

2．字母组数目及组号的修改。使用者可选用 22 种字母组中的任意几组，但一次最多只能选用十组；

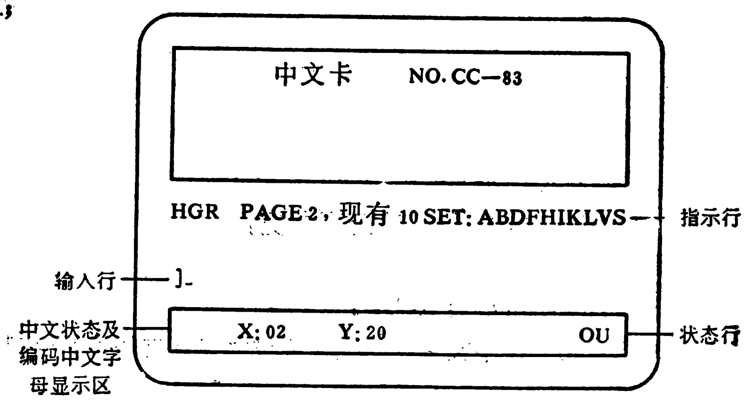


图　6-1-1

3．是否成为(TURNKEY系统及选择 TURNKEY 的文件名。

系统修改必须在开机进入超级汉卡系统后进行。在中文 BASIC 状态下执行 MODIFY 系统程序，通过修改 IDSTAMP 文件实现系统的修改。

一、超级汉卡系统的启动

超级汉卡通常插在 3 号扩充插座上，把超级汉卡系统软盘置于 1 号软盘驱动器，打开电源后，如果引导成功，将出现如图 6-1-1 所示的字幕。

1．指示行

这行中的 HGR PAGE 2 表示使用高分辨率第 2 页显示中文字幕，并指出现有 SET A，SET B，SET D，SET F，SEF H，SET I，SET K，SET L，STE N，SET S 10组字母组可供选用。

2．状态行

当汉字输入时，这一行左面显示标志中文输入状态的"中文"二个字以及取码时显示相应的中文字母。

这一行的中间部分显示光标的 x，y坐标。屏幕最左方，x 坐标值定义为 1，最右方为 34，屏幕的最上方，y 坐标值定义为 1，最下方为 20。程序装入后，先显示 x = 02，y = 20，光标每移动一个位置，其坐标值作相应修改并显示。

这一行右面第 2 个字符位置显示选用的字母组号(0～9)，初始值为 0 (预定)。

这一行最右面的位置显示英文大、小写状态，初始值为 U(大写)，若处于小写状态，将显显示 L。

3．输入行

若这一行上出现中文 APPLE SOFT BASIC 标记]，并在其后面闪动一个光标，说明启动成功，可以使用超级汉卡系统状态下的中文 BASIC 语言编程序或输入命令。

由于光标闪动，易于辨认。现有汉卡中，光标不会闪动

4．中、英文字幕

在超级汉卡系统中，可显示 20 行，每行 34 个共 680 个英文字母，中文可显示 10 行，每行 17 个字共 170 个汉字。当中、英文混合显示时，系统初始设定显示 20 行字母，一旦有中文要显示时，系统会自动调整显示位置，即多分配一个英文行以显示汉字，如图6-1-2所示。在现有汉卡中，不论显示中文或英文都只能显示 10 行，所以超级汉卡能以有限的屏幕显示更多的信息。

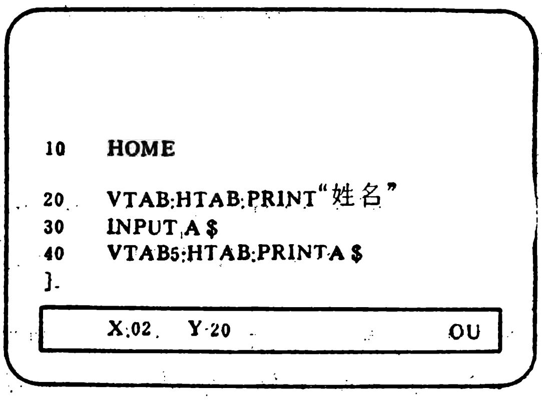


图 6-1-2

在现有汉卡中，END，LEFT$ 以 E，L，显示、打印。在超级汉卡中，它们都以完整的形式显示、打印。

二、修改 IDSTAMP 文件

执行 MODIFY 程序，用户可以根据需要修改 IDSTAMP 文件。修改过程如下：

（1）执行 MODIFY 程序，系统输出如下信息：

```
]RUN MODIFY
HGR FAGE 2，现有10 SET:ABDFHIKLNS
不做 USER TURN KEY SYSTEM
USER TURN KEY FILE NAME:C
HGR PAGE 1 OR 2(1/2)?2
```

并询问，用户准备选择 PG1 或 PG2 作中文字幕显示区。

（2）用户必须以 1 或 2 回答，假设回答 2，接着系统输出：

```
第 0 组及开机预定SET(A-Z)? A
```

询问以哪一组作为系统启动时显示及打印的字体，用户必须以 A～U 中任一字母来回答。

（3）接着系统询问，还要选择几组，用户需要回答选择的组数。系统规定一次最多只能选 10 组，然后用户必须输入各组的选择：

```
要加几组 SET (0-9)? 2
第 1 组 SET = (A-Z)? D
第 2 组 SET = (A-Z)? E
是否做 USER TURN KEY(Y/N)? N
```

（4）在系统输出“是否做 USER TURN KEY(Y/N)?”后，回答 N，系统输出用户本次的选择，并要求用户确认。

```
HGR PAGE 2，现有 3 SET:ADE
不做 USER TURN KEY SYSTEM
USER TURN KEY FILE NAME:C
正确吗?
```

（5）用户可以按任一键确认。然后系统把这次选择写回 IDSTAMP 文件，并装入执行 CCDOS。屏幕上出现如图 6-1-4 所示的字幕。在指示行显示了这项选择的内容：

HGR PAGE 2，现有 3SET：ADE

修改过程结束。

三、22 种字母组的选用

修改 IDSTAMP 文件之后，可以用 CTRL-n 选择已选定的几组字母组中的任一组，其中 n 是组别编号、CTRL-n 可以在直接式中使用，也可以在间接式中使用。

（1）直接式

在修改 IDSTAMP 文件过程结束之后，不论所选择的第 0 组是哪种字母组，键盘输入仍然是英文大写字母，可以输入命令、程序，其输出的字体是选定的第 0 组字体。当用 CTRL-1 指定改变显示的字体之后，键盘输入不再是英文大写字母，输入的命令被拒绝执行，出现

“SYNTAX ERROR”，只有按 RESET 键才能恢复。

（2）间接式

格式：PRINT CHR$(6);″n″

CHR$ 中的 6 是 CTRL-F 所对应的 ASCⅡ码，n 为组别编号，它不可以是变量，只能使用数字。

例：下面的程序是打印 22 种字母组表格的程序。

```
10  POKE 1403, 1:POKE 1659, 0:POKE 1787, 4:POKE 1915,1:POKE 2043,30
20  FOR I=65 TO 86
30  PRINT ″BLOAD SET″;CHR$(I);″, A$8A00″
33  PRINT CHR$(6);″0″;
35  PRINT ″SET″;CHR$(I);
36  PRINT CHR$(6);″1″;
40  HTAB 7:FOR J=32 TO 63:PRINT CHR$(J);:NEXT J:PRINT
45  HTAB 7:FOR J=64 TO 95:PRINT CHR$(J);:NEXT J:PRINT
47  HTAB 7:FOR J=96 TO 125:PRINT CHR$(J);:NEXT J:PRINT
49  PRINT
50  NEXT I
60  POKE 1403, 255
```

其中语句 33 说明表格的表目是以第 0 组字母组打印的，语句 36 说明表格内容是以第 1 组字母组打印的。

第三节　汉字输入及汉字系统存贮器的组织

超级汉卡系统使用的汉字编码及汉字输入法和现有汉卡相同，但操作更简单，它不再使用 CTRL-D 二键，其编辑功能和现有汉卡略有不同。

一、输入操作

1．汉字输入

在超级汉卡系统中只要按 CTRL-L 二键就可以从英文输入状态转换到中文输入状态，并且在状态行的左角显示标志中文输入状态的“中文”二字，而现有汉卡系统中，却需要依次按 CTRL-D，CTRL-L。中文编码输入时，显示相应的中文字母。中文字输入结束可以在中文输入状态下直接按 RETURE 键，而现有汉卡系统中须先按 CTRL-L 回到英文输入状态后，RETURN 功能才有效。

当输入出错时，不论汉字由几码组成，按一次←键就可以回到原汉字输入位置。比现有汉卡操作方便。

2．英文输入

在超级汉卡系统下，英文可以大、小写输入，并可选用22种字母组中的任一组。系统最多选定十组字母备用。

开机时，系统设定英文为大写，状态行右侧显示U字表示现在英文是大写。按 CTRL-A 键可改变英文大、小写的状态，若原来是英文大写，按 CTRL-A 键后，英文变为小写，

状态行显示 L；若原来是小写，按 CTRL-A 键后重新变为大写。

若按 CTRL-K键，不论原来英文是大写或小写一律变为大写，但在输入第二个字母后即自动返回英文小写，以便在专有名词或句首输入时使用。

开机后字母组自动设定为第 0 组。若要使用其它字母组，可按上节所述方法选定。这时状态行上字母组组号显示区显示当前使用的字母组组号，同一行中容许输入多种字母组字体。为了操作方便，打完一行后，按 RETURN 键，字母组回到第 0 组，并回到英文大写状态。

如果要求在输入时，字母组及大、小写状态不受 RETURN 影响，必须修改二个存贮单元中的内容。

（1）38398($95FE)

BIT 7＝1 则 RETURN 后字母组状态不变；

BIT 7＝0 则 RETURN 后取 BIT0～BIT3 作为字母组状态。

（2）38399($95FF)：

BIT 7＝0 则 RETURN 后固定回到大写状态；

BIT 7＝1 则 RETURN 后大小写状态不受影响。

以上二个单元中内容在开机时设定为 0，修改使用 POKE 语句。RESET 或由英文进入中文状态都不再设定。

例如：POKE 38398,2 RETURN 后，状态行的字母组号码变为 2，且状态行的字变成字母组 2 的字母；

POKE 38398,129 RETURN 后，状态行的字母组号码不变，但状态行的字变成字母组 1 的字母；

POKE 38399,0 RETURN 后回到大写状态；

POKE 38399，255 RETURN 后大小写状态不改变。

二、编辑功能

本节指出超级汉卡系统和现有汉卡系统中编辑功能的相异处。

在超级汉卡中 ESC 的功能和现有汉卡中相似，但在超级汉卡系统中，光标的左、右移动以一个英文字符为单位，光标的上、下移动是以一个英文显示行为单位，当光标通过一个包含有汉字的显示行，必须接连击键二次。而在现有汉卡系统中仅需一次。在超级汉卡系统中执行<ESC>E 及<ESC>F 功能时是以一个英文显示行作为一行，如果光标所在行包含有中文字，将会在屏幕上遗留下汉字的上或下半部，虽然不影响信息内容，但达不到美化屏幕的目的。

在现有汉卡中按住→或←键，可实现光标连续移位，而在超级汉卡中要和重复键配合，且以一个英文字符为单位移动。

在现有汉卡系统中，CTRL-H 可能抹去一个汉字，也可能抹去半个汉字，视这个汉字的中文字母编码取码多少而定。在超级汉卡中，不论取码多少，总是抹去一个汉字。

三、超级汉卡系统与其它系统间的相互转换

超级汉卡可以工作于现有汉卡系统，也可工作于超级汉卡系统，用除了 7 个指令之外的语句编写的英文 BASIC 程序也可以在超级汉卡系统下运行，超级汉卡运行过程中使用存贮

域的情况，可在进入监控系统以后了解它。下面将介绍这些系统间的相互转换。

1．超级汉卡系统与现有汉卡系统间的转换

（1）在超级汉卡系统中，中文 BASIC 提示符 ］后输入 PR#3 命令可以进入现有汉卡系统。系统重新设置 HIMEN 为 $8FFF，支持它的是 DOS3.3 操作系统。

（2）由现有汉卡系统进入超级汉卡系统，先按 RESET 键（这里不可同时按 CTRL 与 RESET 键）、然后输入命令 BLOAD RET CCDOS 把机器语言文件 RET CCDOS 装入到内存，最后输入字符&，进入超级汉卡系统。

2．超级汉卡 BASIC 系统与英文 BASIC 系统间的转换

（1）由超级汉卡 BASIC 系统进入英文 BASIC 系统

输入命令：CALL 64098 或 TEXT

由现有汉卡 BASIC 系统进入英文 BASIC 系统还可以使用 CTRL-RESET，而这里 CTRL-RESET 不起作用，仍保持在超级汉卡系统状态。

（2）由英文 BASIC 系统进入超级汉卡 BASIC 系统

输入命令：RUN CCDOS或 & RETURN

3．超级汉卡系统与监控系统间的相互转换

（1）由超级汉卡系统进入监控系统使用

CALL-151 命令

（2）由监控系统进入超级汉卡系统使用

CTRL-C，CTRL-B 命令

四、超级汉卡系统存贮器的组织

超级汉卡系统使用存贮器的情况如图 6-3-1 所示。HIMEM 取决于用户选择的字母组数其关系如下表：

字母组数	HIMEM
1组	$8D00
2组	$8A00
3组	$8700
4组	$8400
5组	$8100
6组	$7E00
7组	$7B00
8组	$7800
9组	$7500
10组	$7200

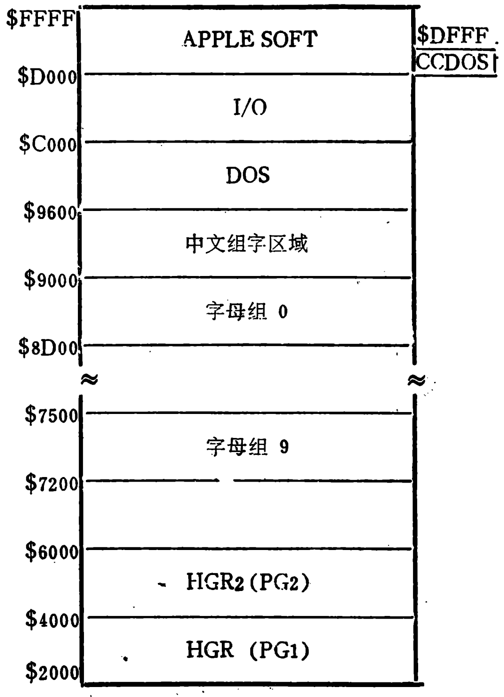


图 6-3-1

进入监控系统，由 0 页 $67～$74 单元中的内容可以了解中文 BASIC 程序使用存贮器的情况。见第五章第二节叙述。

当使用 PG1 或 PG2 作为中文字幕显示区时，可使用的存贮区如图 6-3-2 所示。

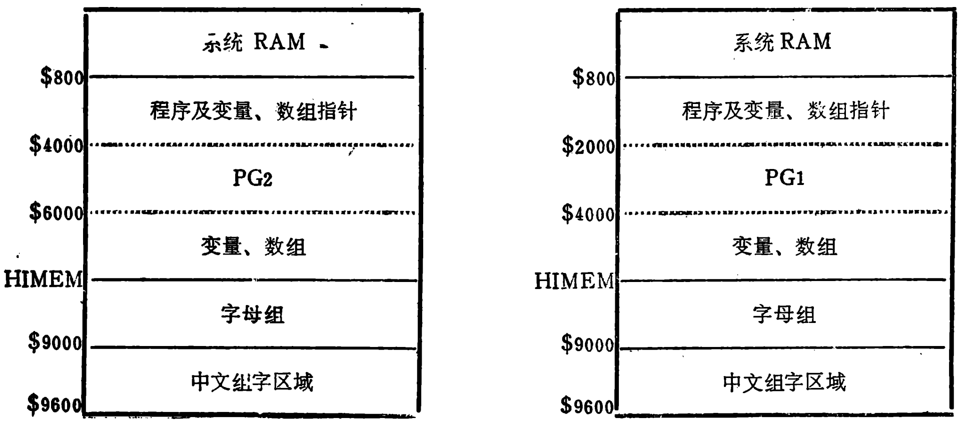


图 6-3-2

第四节 中文 APPLE SOFT BASIC 语言

在超级汉卡系统中，不能使用的英文 BASIC 语句仅 7 个，超级汉卡系统特殊功能语句和现有汉卡系统部分兼容，扩充的&语句可以直接用在程序中，实现从英文 BASIC 系统到超级汉卡 BASIC 系统的转换。

一、中文 BASIC 特殊功能语句

1. CALL 语句

在超级汉卡系统中提供三条 CALL 语句对屏幕进行编辑

(1) CALL 38356 相当于 ESC-E

清除从光标位置到光标所在行行末的字幕；

(2) CALL 38365 相当于 ESC-F

清除光标所在行的整行字幕；

(3) CALL 38374

清除从光标位置到屏幕右下角的整个屏幕内容。

现有汉卡中清除整个屏幕的 CALL 49941 语句在超级汉卡中不再使用，而以HOME语句实现这一功能。

2. 控制打印输出的 POKE 语句

除了关闭打印机语句必须以 POKE 1403，255 代替现有汉卡中的 POKE 1403，0 语句以及开机设定每行打印的中文字字母数不同外，超级汉卡中控制打印输出的 POKE 语句均与现有汉卡中的对应语句兼容。

假设汉卡插在 3 号扩充插座，打印机控制卡插在 1 号扩充插座。则：

(1) POKE 1403，1 打开打印机；

(2) POKE 1403，255 关闭打印机；

(3) POKE 1659, 0　　大字横印；
(4) POKE 1659, 1　　大字直印；
(5) POKE 1659, 2　　小字横印；
(6) POKE 1659, 3　　小字直印；
(7) POKE 1787, N　　设定行与行间隔，开机设定 N=4；
(8) POKE 1915, N　　设定英文字与英文字间隙，开机设定 N=0；
(9) POKE 2043, N　　设定每行中文字字数，开机设定 N=30；
(10) POKE 1531, 12　　自动跳页功能。

控制自动跳页功能需使用语句 CALL 38347。

现有汉卡系统中的光标定位功能语句 POKE 214，H 及 POKE 215，V 在超级汉卡系统中以与英文 BASIC兼容的 HTAB、VTAB 语句取代。

在汉字输入时不再使用 CTRL-D 二键，现有汉卡系统中控制屏幕左下角中文状态及中文字母显示与否的 POKE 253，0 和 POKE 253，255 语句在超级汉卡系统中不再使用。

二、超级汉卡中扩充的和英文 BASIC 兼容的函数及语句

1．POS 语句

它确定当前光标所在的垂直座标值。

例如：]?"日月"；POS(0)

命令执行后，屏幕上显示

　　日月 4

2．SPC 语句

其功能是跳过一个空白位置。它只能用在 PRINT 语句之中。

格式：SPC(n)

n：要跳过的空白位置数目。

例：把一个标题显示在屏幕中间。

```
10 HOME
20 PRINT SPC(11);"学生成绩表"
30 END
```

屏幕上一行可以显示17个汉字，标题为 5 个汉字，所以标题二边各留相当于 6 个汉字的空白位置，取 n=11。

3．TAB 语句

这是一个自定义打印格式函数，它可以在一行中指定显示的位置及打印的位置。

格式：TAB(n)

n 可以是变量，也可以是常数，n 的范围是 0～255。它只能用在PRINT 语句中。

例：试把标题“学生成绩表”及栏标题“学号、姓名、成绩”显示在屏幕上部正中间。

```
10 HOME
20 PRINT TAB(11);"学生成绩表"
30 PRINT TAB(3);"学号"; TAB(13);"姓名"; TAB(23);"成绩"
40 END
```

本例中，每栏占 4 个汉字位置，每两栏之间留一个汉字位置的空格，表格两边各留一个

半汉字位置。

4. HTAB、VTAB语句

HTAB 是水平定向语句，VTAB 是垂直定向语句，HTAB 语句和 VTAB 语句配合使用，可把光标移到屏幕的任一个位置。

格式：HTAB n

n：水平位置，范围是 1～34。

VTAB n

n：垂直位置，范围是 1～20。

例：试把“超级汉卡”四个字显示在屏幕正中间。

```
10  HOME
20  VTAB 11:HTAB12:PRINT "超级汉卡"
30  END
```

5. INVERSE，NORMAL语句

显示格式语句。

格式：INVERSE

使字符以白底黑字显示。

NORMAL

使字符显示恢复到正常状态，即黑底白字。

这两个语句没有其它参数，也不影响键盘打进去的字符和已显示在屏幕上的字符，仅对后来的 PRINT 语句中输出的字符起作用。

例：上例中，试将“超级汉卡”四个字以白底黑字显示后，重新恢复正常显示状态。

```
10  HOME
20  VTAB11:HTAB12:INVERSE:PRINT "超级汉卡":NORMAL
30  END
```

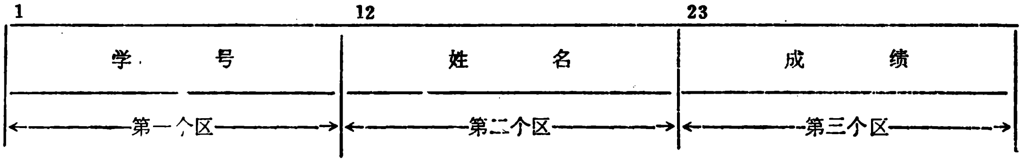
6. PRINT 语句

PRINT 语句及在 PRINT 中使用分号的功能不再说明。PRINT语句中使用逗号将信息显示在一行中固定的三个区域中。这三个区域的起始位置分别是 1，12，23。

例：

```
10  A$ = "学号":B$ = "姓名":C$ = "成绩"
20  PRINT A$, B$, C$
30  END
```

执行结果见图 6-6-1。



6-6-1

7. TEXT 语句

TEXT 语句没有其它参数，执行这个语句实现从超级汉卡系统到英文 BASIC 系统的转换。CALL 64098 也能实现这个转换，但是 CALL 64098 语句使用在程序中时，当返回英文 BASIC 后，不再继续执行下面的英文 BASIC 程序，使用TEXT可以实现这一功能。

三、扩充的＆语句

＆语句实现从英文 BASIC 系统重回超级汉卡 BASIC 系统的功能。＆语句也没有其它参数。

例：设计一个程序，先在屏幕正中以反相显示“超级汉卡”，接着转入英文 BASIC 系统显示第二个屏幕页，其左上角分别以正常显示方式，闪烁显示方式及反相显示方式（白底黑字)显示“S.C.C.G-83”，第三个屏幕页显示 APPLE Ⅱ 低分辨率的绘图功能，第四个屏幕页显示APPLE Ⅱ 高分辨率第一页的绘图功能，最后重回超级汉卡系统，使用 PG2在第五个屏幕页显示“中文信息处理”。

有关 APPLE Ⅱ 的绘图语句，请参阅第七章。

程序设计如下：

调用系统盘的 MODIFY 文件以修改 IDSTAMP，选择 PG2 作为中文字幕显示区。在超级汉卡系统 BASIC 提示符 ］后输入程序。

```
10  HOME:VTAB11:HTAB12:INVERSE:PRINT"超级汉卡":NORMAL:GET A$
20  TEXT:HOME:PRINT "S.C.C.G-83":FLASH:PRINT "S.C.C.G-83":
    INVERSE:PRINT "S.C.C.G-83":NORMAL:INPUT A$
30  GR:FOR I=0 TO 39:COLOR=I/2:VLIN 0, 39 AT I:NEXT:GET A$
40  HGR:FOR I=0 TO 279:HCOLOR=I-INTC(I/8)*8:HPLUT 145, 96 TO I,
    O:NEXT:GET A$
45  SPEED=150
50  &:VTAB17:PRINT "中文信息处理"
60  END
```

执行这个程序，语句 10 将闪烁显示“超级汉卡”，语句 20 分别以正常显示，闪烁显示及白底黑字显示状态显示“S.C.C.G-83”，语句 30显示低分辨率的绘图功能，语句 40 显示高分辨率第一页的绘图功能，语句 50 在高分辨率第二页上显示“中文信息处理”。

在超级汉卡中文 APPLE SOFT 中不可执行的绘图语句，可以在 GR 前面加 TEXT 语句，先返回英文系统，再执行绘图语句，重新进入中文系统只要执行＆语句。

在超级汉卡中文系统中不可执行 FLASH 语句，如果执行这个命令，出现在屏幕的字不正确，复原办法有二个：

（1）按 TEXT 进入英文系统，输入 NORMAL 后，再按＆回到中文系统；

（2）执行 POKE 243，0。

第七章　APPLE Ⅱ的绘图功能

APPLE Ⅱ 微机具有绘图功能，在 APPLE SOFT BASIC 语言中提供了绘图语句，利用这些绘图语句编制的绘图程序，可以在屏幕上画出两种图形：低分辨率图形和高分辨率图形。后者分辨率高，比较常用。在主机内存 RAM 中有两页存贮区(每页 8K 字节)作为高分辨率图形显示缓冲区，分别称为高分辨率图形第一页和高分辨率图形第二页。配有汉卡的主机，使用高分辨率图形第二页作为汉字字幕显示缓冲区，这样，在汉字系统下，可以在图形显示中配上汉字说明。在高分辨率图形显示状态下，APPLE Ⅱ还提供了另一种绘图方法，即高分辨率图形表。它以绘图向量序列描述绘出一个图形的全过程，每个图形的绘图向量序列，称为图形定义，存贮在主存中，并与它们的索引组成一个图形表。一个图形表中最多允许包含 255 个图形的绘图向量，它可以保存在磁盘或磁带上。APPLE SOFT 语言中提供了四个处理图形的语句，使用这些语句编制的程序可以调用图形表中的任一个图形的绘图向量序列，绘出或消除一个图形或改变输出图形的大小、方向。采用这种方法，可以把汉字看作图形，从而建立汉字图形表，这样，可以在不配备汉卡的主机中为显示的图形配注汉字。

第一节　低分辨率图形

一、低分辨率图形显示格式

低分辨率图形显示时，APPLE Ⅱ使用二页存贮区(每页 1K 字节)作为低分辨率图形显示缓冲区，分别称为低分辨率图形第一页和低分辨率图形第二页。它们占用的存贮区地址分别为 \$400～\$7FF，\$800～\$BFF。每页可以在屏幕上显示 48行×40 列。行与列的每一个交点在屏幕上显示为一个小的长方形，共可显示 1920 个长方形。见图 7-1-1 所示。每个长方形的色彩变化共有 16 种可供选择。

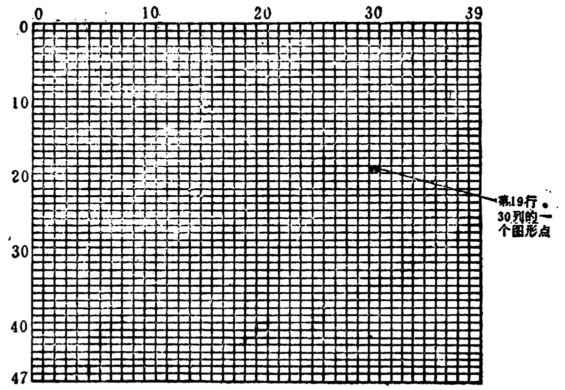


图 7-1-1

二、低分辨率图形显示语句

1．进入低分辨率图形方式语句 GR

格式：GR

执行这个语句后，屏幕上部40行×40列位置全部清除成黑色，将作为低分辨率图形显示区，并保留了下面4行作为文字显示。

执行GR命令后，要把屏幕转变成全屏幕的图形显示(40列×48行)，可以使用下面的命令：

POKE－16302，0

在全屏幕图形显示时，如果要重新显示出下面4行的文字，可以使用GR命令，在显示文字的同时清除图形显示，如果要保留前面40行的图形，同时把文字重新显示出来，可以使用命令：

POKE－16301，0

2．返回文字显示方式语句TEXT

格式：TEXT

执行这个语句后，屏幕显示由低分辨率图形方式返回文字显示方式。由于二者显示方式使用相同的主存缓冲区，而且执行TEXT语句时，不清除这个缓冲区的内容，所以，APPLE Ⅱ将图形资料转变成文字资料，从而使屏幕上出现奇怪的文字，使用ESC－@或HOME命令可以清除屏幕显示。

3．颜色选择语句COLOR

在低分辨率图形显示方式下，可以使用COLOR语句选择不同的绘图颜色。

格式：COLOR＝颜色代码

颜色代码与显示的颜色对应关系如下表。

颜色代码	颜　色	颜色代码	颜　色
0	黑色	8	棕色
1	深红色	9	橙色
2	深蓝色	10	灰色
3	紫红色	11	粉红色
4	深绿色	12	绿色
5	灰色	13	黄色
6	中蓝色	14	浅绿
7	浅蓝色	15	白色

在GR语句执行后，自动设定COLOR＝0，即为黑色(无显示)，绘图之前，需用这个语句选定绘图的颜色。例如，COLOR＝13，则可将颜色选定为黄色。

4．绘点语句PLOT

格式：PLOT A，B

A、B为表达式，A给出所绘点的水平座标值，B给出了它的垂直座标值。

这个语句给出行与列交叉的一个小方块。

例1

```
10 HOME
20 GR
30 COLOR＝13
40 PLOT 6，16
```

这个程序在屏幕的第16行，第6列相交处，显示一个黄色的小方块，见图7-1-2所示。

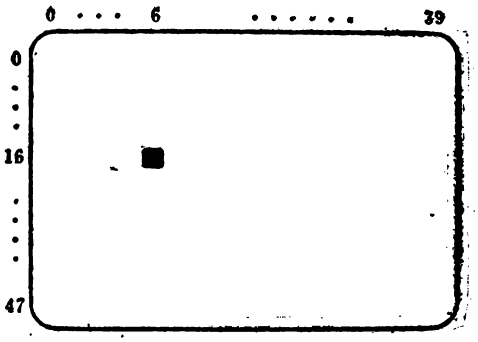


图 7-1-2

例 2

```
10 HOME
20 GR
30 COLOR = 13
40 FOR I = 0 TO 39
50 PLOT  I, I
60 FOR J = I TO 100:NEXT J
70 NEXT I
```

这个程序绘出一条由左而右的斜线方块，如图 7-1-3 所示，其颜色为黄色。其中语句 60 延迟绘图速度。

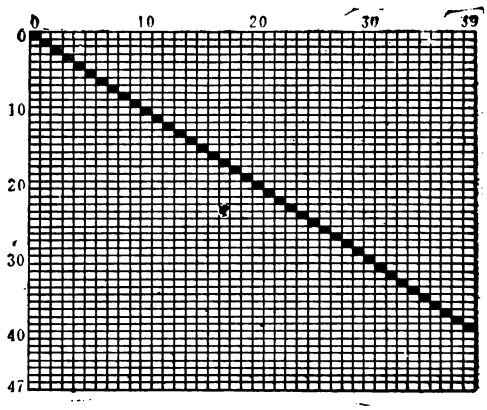


图 7-1-3

5．绘垂直线语句 VLIN

执行 VLIN 语句可以在屏幕上用所选定的颜色绘出一条垂直的直线。

格式：VLIN A，B AT C

A，B，C 为表达式。

A：所画垂直线的起始座标值；

B：所画垂直线的终点座标值；

C：所画垂直线的水平座标值；

A，B 的取值范围是 0～47，而且 B 必须大于 A，C 取值范围是 0～39。

例 3　将屏幕第 8 列的 41 个小方块全部显示为橙色，程序如下：

```
10 HOME
20 GR
30 COLOR=9
40 VLIN 0, 41 AT 8
```

程序执行的结果见图 7-1-4 所示。

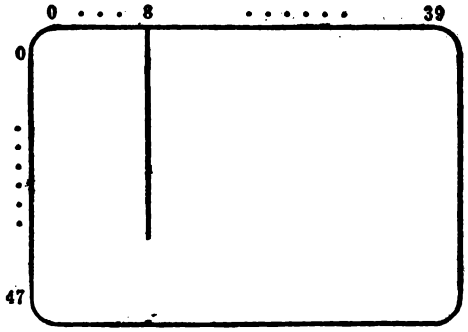


图 7-1-4

6. 绘水平线语句 HLIN

执行 HLIN 语句可以在低分辨率图形页上由左至右绘出不同的直线。

格式：HLIN A，B AT C

A，B，C 为表达式。

A：所画水平线的起点座标值；

B：所画水平线的终点座标值；

C：所画水平线的垂直座标值；

A，B 取值范围 0～39，且B必须大于 A，C 取值范围 0～47。

例 4　将屏幕第 8 行的 40 个小方块全部显示为蓝色的图形。程序如下：

```
10 HOME
20 GR
30 COLOR=6
40 HLIN 0, 39 AT 8
```

程序执行的结果见图 7-1-5 所示。

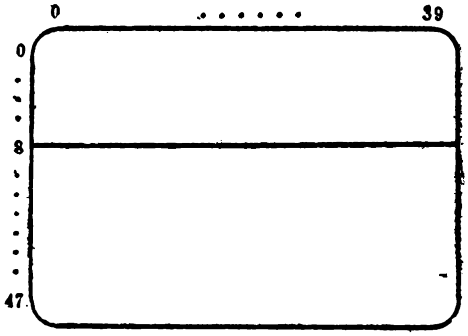


图 7-1-5

7. 检测颜色码语句 SCRN

执行这个语句可以检测出屏幕上某一小方块所显示的颜色的颜色代码。

格式：SCRN(A, B)

A：被检测的小方块的水平座标值；

B：被检测的小方块的垂直座标值。

例 5

```
10 HOME
20 GR
30 COLOR = 14
40 PLOT 12, 12
50 PRINT SCRN(12, 12)
```

执行这个程序后显示颜色代码为 14。

第二节　高分辨率图形

一、高分辨率图形显示格式

APPLE Ⅱ使用二个内存 RAM 缓冲区作为高分辨率图形显示缓冲区，分别称高分辨率图形第一页和高分辨率图形第二页，它们使用内存 \$2000～\$3FFF 和 \$4000～\$5FFF 的存贮区，每页占用 8K 字节。

高分辨率图形显示时，屏幕的水平部分为 280 个点，垂直部分分成 192 个点，一共显示出 53760 个点。座标原点在屏幕的左上角，如图 7-2-1 所示。图形的颜色共有六种可供选择，它们是黑，白、绿、紫、蓝、橙。

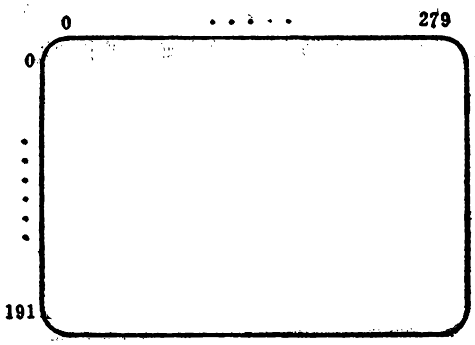


图 7-2-1

屏幕上显示的每一点和 RAM 图形缓冲区中的一位(bit)相对应。一个字节中的 7 位将显示在屏幕上，剩下的 1 位用来选择该字节中其它各位所对应的屏幕上点的颜色。屏幕上的一行对应 40 个字节的缓冲区。

配置汉卡的主机，汉字显示使用高分辨率图形第二页的缓冲区，在汉字系统中，可以在高分辨率图形第二页上绘图，并配注汉字。

二、高分辨率图形语句

APPLE SOFT 提供了在高分辨率图形第一页或第二页上绘图的语句。这些语句是：

1. 进入高分辨率图形方式语句 HGR，HGR 2

进入图形第一页的方式语句格式：HGR

这个语句把屏幕显示设定为高分辨率图形第一页，屏幕上方 160×280 点阵用于显示图形，屏幕下部 32×280 点阵用于显示 4 行文字幕。执行这个语句，屏幕上图形区清除成黑色，如图 7-2-2 所示。若使用命令 POKE－16302，0，可使图形、文字混合显示转变为全屏幕图形显示。

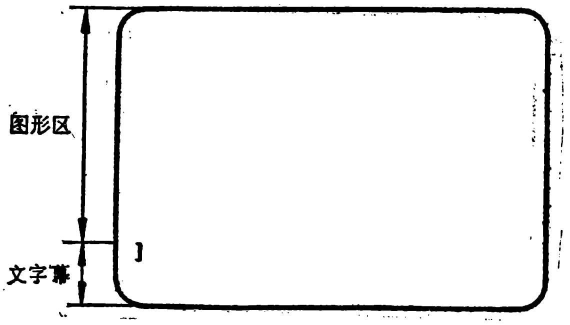


图 7-2-2

进入图形第二页的方式语句格式：HGR2

这个语句把屏幕显示设定为高分辨率图形第二页，全屏幕显示图形，不显示文字幕。

但执行这个语句之后，若使用命令：POKE－16301，0，可使全屏幕图形显示转变为图形、文字混合显示。

2．颜色选择语句 HCOLOR

使用 HGR 或 HGR2 进入高分辨率图形显示方式，屏幕上图形区呈现黑色。在绘图之前需用 HCOLOR 语句选择绘图颜色。

格式：HCOLOR＝颜色代码

颜色代码与对应的显示颜色如下表：

颜色代码	显示颜色	颜色代码	显示颜色
0	黑	4	黑
1	绿	5	橙
2	紫	6	蓝
3	白	7	白

3．绘图语句 HPLOT

绘图语句 HPLOT 有下面两种格式：

（1）绘点格式

格式：HPLOT A，B

A，B 为表达式。A 是绘点的水平座标值，B 是绘点的垂直座标值。

例 6　设计一个程序，使屏幕上第 1 行、第 1 列，第 1 行、第 279 列，第 191 行、第 279 列，第 191 行、第 1 列的四个点全部显示为绿色。程序如下：

10 HOME

20 HGR

30 HCOLOR＝1：POKE－16302，0

40 HPLOT 1，1

50 HPLOT 279，1

60 HPLOT 279，191

70 HPLOT 1，191

执行这个程序后，屏幕显示如图 7-2-3 所示。

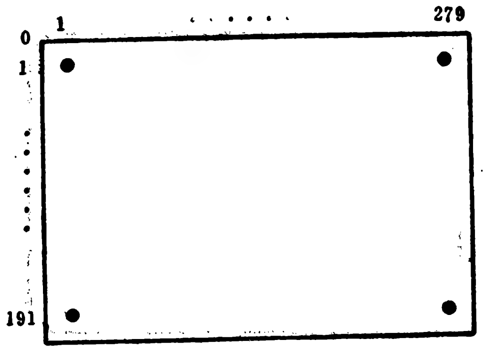


图 7-2-3

（2）绘线格式

格式 1：HPLOT A，B TO C，D

其中 A，B，C，D 为表达式。

A，B 分别为直线起点的水平和垂直座标值；

C，D 分别为直线终点的水平和垂直座标值。

这种格式绘出一条直线。

例 7　编制一个程序，在屏幕上用蓝色绘出由左上角到右下角的一条对角线。程序如下：

10 HOME

20 HGR

30 HCOLOR = 6 :POKE - 16302，0

40 HPLOT 0，0 TO 279，191

程序执行的结果如图 7-2-4 所示。

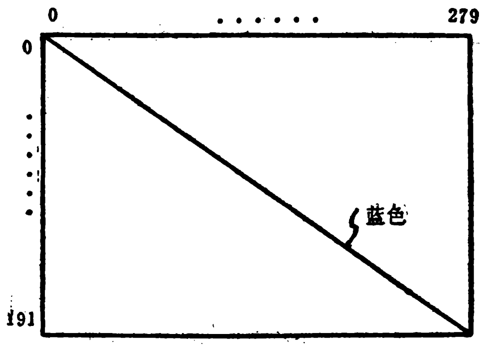


图 7-2-4

格式 2：　HPLOT A，B　TO　C，D　TO　E，F TO……

这种格式可以绘出折线。

其中 A，B，C，D，E，F，……均为折点处的坐标点。

例 8　在高分辨率第 2 页上画一个长方形，要求从第 8 行、第 31 列开始依顺时针方向绘出水平边长为 216 个点，垂直边长为 80 个点的绿色长方形。程序如下：

```
10 HOME:HGR2:HCOLOR=1
20 HPLOT 31, 8 TO 247, 8
30 HPLOT 247, 8 TO247, 88
40 HPLOT 247, 88 TO 31, 88
50 HPLOT 31, 88 TO 31, 8
```

程序执行的结果如图 7-2-5 所示。

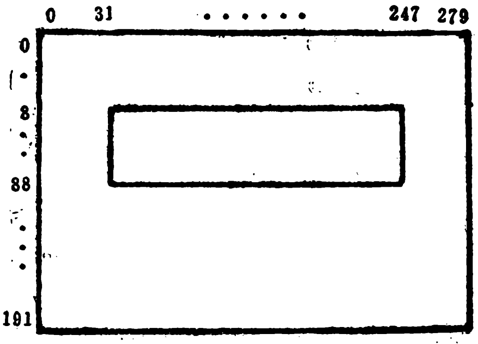


图 7-2-5

程序中语句 20～50 可以用一条语句代替，即：

```
20 HPLOT 31, 8 TO 247, 8 TO 247, 88 TO 31, 88 TO 31, 8
```

结果相同。

为使程序具有更大的通用性，可以在 HPLOT 语句中使用表达式。

例 9　从第 20 行、第10列开始依顺时针方向绘出一个边长为 51 个点的蓝色正方形，程序如下：

```
10 HOME:HGR:HCOLOR=6
20 R=20:C=10:W=51:N=51
30 HPLOT C, R TO C+W-1, R  TO  C+W-1, R+N-1 TO
          C, R+N-1  TO  C, R
```

其中 R、C 为起点的水平、垂直座标值，W、N 分别为水平方向边长及垂直方向边长。如果要更改图形的起绘点或宽度、高度，只要修改语句 20 中相应的变量值。

4．返回文本显示方式语句 TEXT

格式：TEXT

这个语句执行后，从高分辨率图形显示方式返回文字显示方式。

三、绘图实例

高分辨率图形有较高的分辨率，可以绘出各种比较精细的图形和曲线。下面给出两个绘图实例。

例 10　在屏幕上绘出 8 个白色同心圆，圆心在屏幕的第 95 行、第 95 列。程序如下：

```
10 HOME
20 HGR2
```

```
30 HCOLOR=7
40 R1=10:R2=80:S=10:CX=95:CY=95
50 FOR R=R1 TO R2 STEP S
60 FOR X=-R TO R
70 Y=INT(SQR(R∧2-X∧2))
80 HPLOT CX+X, CY-Y
90 HPLOT CX+X, CY+Y
100 NEXT X
110 NEXT R
```

语句 40 中，R1 为同心圆中的最小半径；
R2 为同心圆中的最大半径；
S 为半径的增量；
CX 为圆心的 X 轴坐标；
CY 为圆心的 Y 轴坐标。

循环语句 50 规定从半径最小的圆画起直到画出半径最大的圆。

语句 60 规定绘图时以绘点方式从左向右绘出。

由于圆上各点座标满足 $x^2+y^2=R^2$，所以 $y=\pm\sqrt{R^2-x^2}$。又屏幕上点的座标值均为整数，所以在语句 70 中对 y 值取整。

语句 80 绘出上半圆，语句 90 绘出下半圆。

例 11　绘制一条正弦曲线，曲线的一个周期对应 260 个点，曲线的峰值对应 60 个点。座标原点设在第 90 行、第 10 例。程序如下：

```
10 HGR2
20 HCOLOR=1
30 HPLOT 0, 90 TO 279, 90
40 HPLOT 10, 0 TO 10, 159
50 HCOLOR=5
60 FOR I=0 TO 260
70 HPLOT I+10, 90-60*SIN(I*3.1416/130)
80 NEXT
```

其中语句 30 画出水平座标轴，语句 40 画出垂直座标轴，颜色由语句 20 设定为绿色。语句 60～80 以绘点方式画出正弦曲线，颜色由语句 50 设定为橙色。

HPLOT 中的座标值若为小数，系统会自动取整。

四、在图形中配注汉字

当 APPLE Ⅱ 配备汉卡时，在进入汉字系统后，可以使用高分辨率图形第二页的绘图语句编制绘图程序，在高分辨率图形第二页上绘制图表、曲线，并用中文 APPLE SOFT BASIC 语句输出汉字，在图形上配注汉字等。

例 12　解放以来，我国国民经济稳步增长，国家财政收入逐年提高，可用图 7-2-6 说明。

试设计一个程序，在屏幕上显示这个图表及汉字说明。假设已进入汉字系统，程序如下：

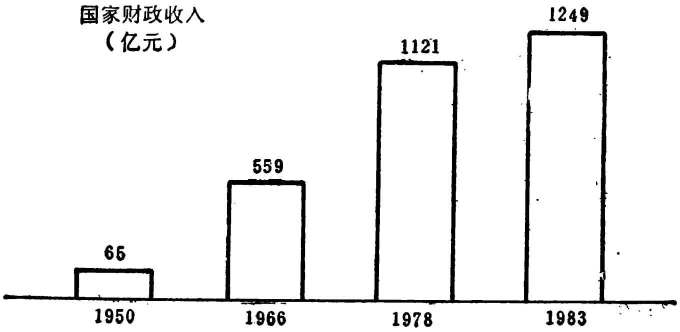


图 7-2-6

```
10  HGR2:HCOLOR=2
20  A=65:B=559:C=1121:D=1249
30  L=160:M=50:N=82:H=L-INT(A/15):GOSUB 180
40  M=98:N=130:H=L-INT(B/15):GOSUB 180
50  M=146:N=178:H=L-INT(C/15):GOSUB 180
60  M=194:N=226:H=L-INT(D/15):GOSUB 180
70  HCOLOR=5
80  HPLOT 0, L TO 279, L
90  POKE 215, 9:POKE 214, 6
100 PRINT "1950";"␣";"1966";"␣";"1978";"␣";"1983"
110 POKE 215, 2:POKE 214, 24:PRINT "1249":CALL 49944
120 POKE 215, 2:POKE 214, 18:PRINT "1121":CALL 49944
130 POKE 215, 4:POKE 214, 12:PRINT "559";CALL 49944
140 POKE 215, 6:POKE 214, 7:PRINT "65":CALL 49944
150 POKE 215, 1:POKE 214, 4:PRINT "国家财政收入":CALL 49944
160 POKE 214, 7:PRINT "(亿元)":CALL 49944
170 POKE 215, 9:E
180 FOR I=M TO N-1
190 HPLOT I, L TO I, H TO  I+1, H  TO  I+1, L
200 NEXT I
210 RETURN
```

语句20把 4 个年份的财政收入赋值给变量 A、B、C、D，语句 30～60 调用语句 180～210 的绘图表子程序，从左到右绘出四个比例图形，颜色为紫色，色彩填满每个比例图形；

语句 80 绘出下部水平线，颜色为橙色；

语句 90～100 在水平线下部配注年份；

语句 110～140 在每个比例图形上方配注当年财政收入的数目；

语句 150～160 在图表左上方配注汉字，对图表作说明。

五、颜色的限制

在高分辨率图形显示方式时，如果使用黑白电视屏幕，若屏幕上每一点对应的内存显示缓冲区中的位(bit)是 1，则显示白点；是 0，则显示出黑点。如果采用彩色屏幕，若缓冲区

中一位(bit)是 0，那么屏幕上相对应的点依然显示黑色，是 1，那么它的颜色跟它在屏幕上的位置有关。也就是说，在高分辨率图形显示时，可以用 HCOLOR 语句选择六种不同的彩色，但屏幕上显示的颜色' 受到限制，这个限制如下：

在偶数列上(水平座标为偶数)只能显示紫、蓝或黑色。若用绿、橙色，则显示出黑色。

在奇数列上(垂直座标为奇数)只能显示绿、橙或黑色。若用紫、蓝色，则显示出黑色。

例 13

```
10 HGR
20 HCOLOR=6
30 HPLOT 20，0 TO 20，159
40 HPLOT 61，0 TO 61，159
```

本程序要在屏幕上显示二条垂直线。然而，结果仅显示从座标点(` ` 0)到座标点(20，159)的一条垂直线，另一条垂直线不显示。因为这条垂直线的水平座标是 61，为奇数，而语句20设定的颜色却是蓝色。

第三节　高分辨率图形表

使用上节所述的高分辨率图形语句绘图，当需要图形在屏幕上进行某些处理时，比如，将图形旋转或放大、缩小等，会遇到困难。APPLE Ⅱ 提供了另一种绘图方法，即高分辨率图形表。

在高分辨率图形表绘图方式中定义了 8 个绘图向量，每一个向量描述往上、下、左、右移动或不移动，同时也描述了是否要绘在屏幕上，它是绘出图形的指令码。要绘出一个图形，先选一个绘图起始点，按绘图路径，把绘图的指令码即绘图向量列出来，这个绘图向量序列就描述了一个图形，它以二进制方式存放在内存中，各个图形的绘图向量序列及其索引组成了一个高分辨率图形表。APPLE Ⅱ 允许一个图形表中最多包含 255 个图形的绘图向量。人们可以随时调用图形表中的某一图形的绘图向量在屏幕上绘出这个图形。APPLE SOFT 在高分辨率图形表绘图方式中提供了 5 条命令，用来处理这些图形。

在调用图形表中的绘图向量进行绘图之前，要进入高分辨率图形显示状态(即执行了 HGR 或 HGR2 语句)，并选定颜色。

当建立了一个具有数字、字母、符号及汉字的高分辨率图形表，就可以在绘图程序中调用图形表中这些数字、字母、符号及汉字的绘图向量，在高分辨率图形、图表上配注数值、汉字等。

一、绘图向量

绘图向量用三位二进制码表示，其代码与绘图动作的关系如下表所示：

绘图向量	符号	绘图动作	绘图向量	符号	绘图动作
000	↑	往上移动，不绘图	100	↑	往上移动，绘图
001	→	往右移动，不绘图	101	→	往右移动，绘图
010	↓	往下移动，不绘图	110	↓	往下移动，绘图
011	←	往左移动，不绘图	111	←	往左移动，绘图

一个绘图向量可以表示为：

位	2	1	0
符号	P	M	M

当P＝0 时表示不画出图形；

P＝1 时表示要画出图形。

MM＝00 表示向上移动；

MM＝01 表示向右移动；

MM＝10 表示向下移动；

MM＝11 表示向左移动。

当用一个绘图向量序列描述一个图形时，先画出图形，再以绘图向量图示绘图过程，然后写出绘图向量序列。

例 14　列出图 7-3-1 所示的正方形的绘图向量序列。

设选取左上角的点作为绘图起点，依顺时针方向绘出。图 7-3-2 给出了绘图向量图示。这个正方形的绘图向量序列用符号表示是

→→→→→→→→

而向量序列是：

101　101　101　101　101　101　101　101。

例 15　列出图 7-3-3 所示图形的绘图向量序列。

先把图形画在方格纸上，并假定从图形中间开始画，再画出绘图路径，由于绘图向量只能往上、下或左、右移动，因此只能用 90° 度角来转弯，如图 7-3-4 所示。

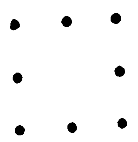


图　7-3-1

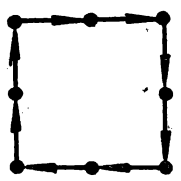


图　7-3-2

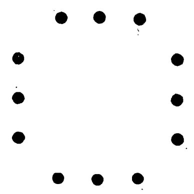


图　7-3-3

接着用绘图向量图示绘画过程，如图 7-3-5 所示。绘图向量序列可用符号表示为

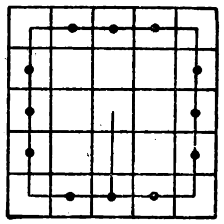


图　7-3-4

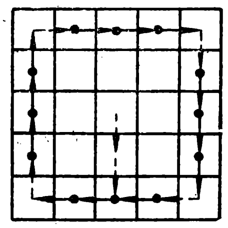


图　7-3-5

↓ ↓ ← ← ↑ ↑ ↑ ↑ → → → → ↑ ↓ ↓ ↓ ← ←

而向量序列是：

010　010　111　111　000　100　100　100　001　101　101　101　010　110　110　110　011　111。

二、绘图向量在内存中的表示方式

APPLE Ⅱ 中用内存的一个字节表示 3 个绘图向量，分成 A、B、C 三个段如下：

C		B			A		
7	6	5	4	3	2	1	0

每一字节的三个绘图向量段规定如下：

1．C 段仅 2 位(bit)，(相当于 P＝0)，所以C段表示的向量只能用于移动，而不会在屏幕上绘点。C＝01 表示右移(←)，C＝10 表示下移(↑)，C＝11 表示左移(←)，但 C＝00无任何作用，即仍在原来的点上。

2．当 C＝00 且 B＝000，则B向量不再具有原来的意义(上移而不绘点)，即 C、B 向量都表示不移动也不绘点。

3．若 A＝000，B＝000，C＝00 表示一个图形的绘图向量序列结束。

根据以上的规定，例 15 中的绘图向量序列在内存中的存贮形式如下表：

字　节	绘图向量图示 C B A	内存存贮形式 A B C	十六进制值
0	↓ ↓	00 010 010	12
1	← ←	00 111 111	3F
2	↑ ↑	00 100 000	20
3	→ ↑ ↑	01 100 100	64
4	→ →	00 101 101	2D
5	↑ →	00 010 101	15
6	↓ ↓	00 101 110	36
7	→ ↓	00 011 110	1E
8	←	00 000 111	07
9		00 000 000	00

三、图形表

图形表包括各个图形的绘图向量序列表格及其索引，它实际上是内存中一个二进制的数据表格。它描述了该表中有多少个图形的绘图向量序列，同时也指出了它们在表中的位置。图形表可以建在内存的任一区域，假设一个图形表存在内存中的起始地址为 S，各个图形的绘图向量序列表格起始地址相对于 S 的位移为 Di，i 表示第 i 个图形。那未图形表的格式如图 7-3-6 所示。

其中图形表的起始地址必须存于 E8，E9 单元。E8 单元存放地址的低位，E9 单元存放地址的高位。

四、建立图形表

建立图形表，首先必需确定这个图形表要定义几个图形；图形表存放的起始地址；各个图形的绘图向量表格的起始地址相对于图形表起始地址的位移以及各个图形的绘图向量序列，然后按上面所述的图形表格式输入内存。图形表数据输入可以用 POKE 命令来完成，也可以在监控系统下完成。

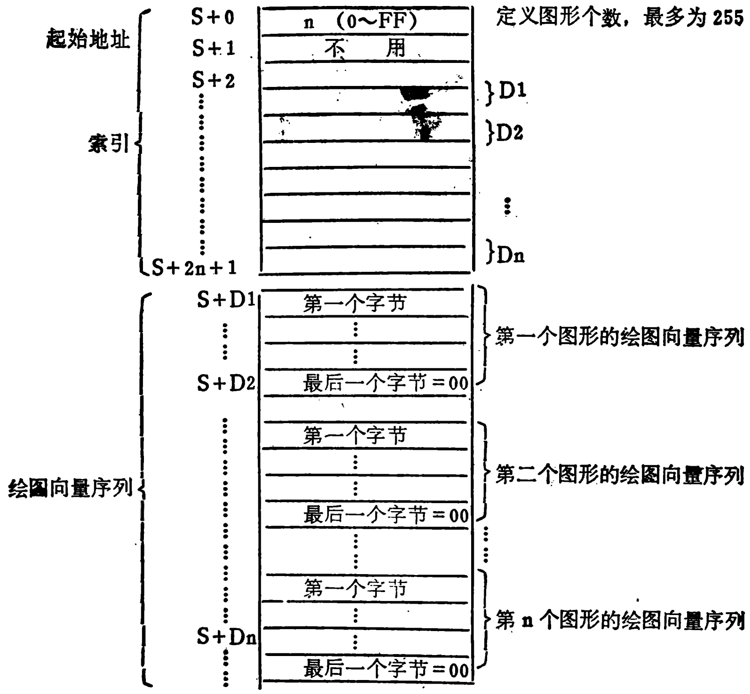


图 7-3-6

例 16　建立一个如例 15 中所示图形的图形表。

设图形表在内存的起始地址为 $6000，由于只有一个图形，$6000 单元中应存入 01，绘图向量序列由 $6008 单元开始存放，$6002 及 $6004 单元中应存入位移量 0008，并设在监控系统下输入，输入过程如下：

```
]CALL－151
*6000:01  00  08  00  00  00  00  00  RETURN
*6008:12  3F  20  64  2D  15  36  1E  RETURN
*6010:07 00 RETURN
```

输入之后，给出如下命令检查输入是否正确

```
*6000.6018
```

再把图形表的起始地址存入 E8、E9 单元

```
*E8:00 60 RETURN
```

检查输入正确后用 CTRL-C 退出监控系统，图形表建立后，就可使用。

通常都把建好的图形表用以下的 DOS 命令保存在磁盘上，设文件名为 SHAPE，图形表占用存贮区的长度为 18 个字节。

```
BSAVE SHAPE, A$6000, L20
```

若要使用这个图形表，须先把它装入到内存，再在 E8，E9 单元中设置图形表的起始地址

```
10  PRINT CHR$(4);"BLOAD SHAPE"
20  POKE 232, 0:POKE 233, 96
```

注意语句 20 中的值均为十进制的值。

综上所述，建立一个图形表的步骤如下：

(1) 先在方格纸上绘出要画的图形；

(2) 确定绘图的起点，设计绘图路径，并给出绘图向量图示；

(3) 给出绘图向量序列在内存中存贮的十六进制形式；

(4) 输入 CALL-151 命令，进入监控系统，确定图形表在内存的起始地址，并按上面所述的图形表格式逐一输入索引及各个图形的绘图向量；

(5) 退出监控系统，并输入 BSAVE 命令把图形表保存到磁盘上。

五、图形表的调用

在高分辨率图形表方式中，APPLE SOFT 提供四个特殊语句，用以调用表中图形的绘图向量进行绘图，并可对图形作旋转，放大或缩小，清除等处理。

1. 调用语句 DRAW

格式(A)：DRAW A AT X，Y

或 格式(B)：DRAW A

A，X，Y 为表达式。

A 指出所绘的图形在图形表中的序号，A 值在 0～255 之间；

B、C 指出所绘图形起始点的水平、垂直座标值，其值分别在 0～279，0～191 之间。

如果用格式(B)，即不指明 B、C，则以最近执行HPLOT，DRAW 或 XDRAW(见下面叙述)的位置作为起始位置，画出指定的图形。

在 DRAW 执行前由 HGR 或 HGR2 语句进入高分辨率图形方式，由 HCOLOR 语句指定图形的颜色，并由下面将要介绍的语句 ROT，SCALE 语句指明旋转的角度及尺寸等。

2. 调用语句 XDRAW

语句格式与 DRAW 相同，即：

XDRAW A AT X，Y

XDRAW A

执行 XDRAW 语句时，是以现有颜色的补色画出图形。所谓颜色互补指的是将两种颜色混合后，显示为黑色。下面几对颜色是互补的：黑与白；蓝与绿；橙与紫。

XDRAW 语句提供了一个简单的清除图形的方法，即当用 DRAW 语句画出一个图形后，再执行一个 XDRAW 语句，而且座标上旋转向量及大小都与画原图形时相同，则将在原图形上以补色重画，即擦掉原图形，但不擦掉原图形以外的图形。

3. 图形比例语句 SCALE

SCALE 语句设定要画出的图形的尺寸。

格式：SCALE = 表达式

表达式取值范围 0～255。

SCALE = 1， 设定每一个绘图向量的范围为一个点，即按原图形的尺寸画图；

SCALE = 2， 设定每一个绘图向量的范围为 2 个点，即画出的图形是原图形的 2 倍；

⋮

SCALE = 255 表示把原图形放大 255 倍；

SCALE = 0 表示把原图形放大 256 倍。

4. 图形旋转语句 ROT

ROT 语句设定绘出图形的旋转角度。

格式：ROT＝表达式

ROT 中表达式的值在 0～255 之间，但最多只有 64 种可能的旋转系数，即0～63，旋转角度由 SCALE 语句与 ROT 语句共同定义。

当 SCALE＝1 时，ROT 只能依一次旋转 90° 的方式画出图形，ROT 值为 0，16，32 48 时对应的旋转角度为 0°，90°，180°，270°；

当 SCALE＝2 时，ROT 的值为 0，8，16，24，32，40，48，56，64 时对应的旋转角度为 0°，45°，90°，135°，180°，255°，270°，315°，360°。

以此类推，当 SCALE＝5 或大于 5 时，最多可达 64 种旋转角度。

如果给定的 ROT 值不是 ROT 语句中规定的旋转系数值，APPLE SOFT 会自动把给定的 ROT 值转换成比它小的规定的旋转系数值。如果给定的 ROT 值大于 64，则以此值除以 64 的余数作为旋转系数。

例 17 编制一个程序，每次画出例 15 中的图形并旋转，重复 50 次，每次重复时都把图形放大。程序如下：

```
10 HGR
20 HCOLOR＝3
30 FOR R＝1 TO 50
40 ROT＝R
50 SCALE＝R
60 DRAW 1 AT 139，79
70 NEXT R
```

程序执行之前，假设图形表已在内存，其起始地址已存放在 E8，E9 单元。

如果每次画出图形后要暂停片刻并擦去图形，可加上以下二个语句：

```
62 FOR I＝0 TO 1000：NEXT I
64 XDRAW 1 AT 139，79。
```

六、汉字图形表

采用高分辨率图形表的绘图方法，可以把一个个汉字作为一个个图形，从而建立汉字图形表，这样不需要硬件汉卡，就可以在绘制的曲线、图表上配注汉字。甚至花边、图案的设计。

例 18 建立一个图形表，它包括数字 0，1，…9，小数点，汉字："布"，"亿"，"米"，符号"("和")"等 16 个图形的绘图向量。

下面以数字"2"字形及汉字"布"字形为例，设计绘图向量。

数字 0，1，…9 及小数点采用 4×6 点阵，汉字及符号"("和")"采用 16×16点阵。

数字"2"字形点阵如图7-3-7所示，绘图向量示于图 7-3-8 中，绘图向量在内存中的存贮形式如下：

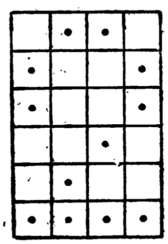


图 7-3-7

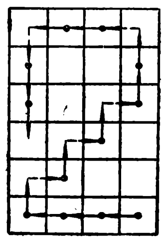


图 7-3-8

字 节	绘 图 向 量	内存存取形式	十六进制值
	C B A	C B A	
0	← ←	00 111 111	3F
1	→ ↑ ←	01 100 111	67
2	→ ↑	00 001 100	0C
3	→ ↑	00 001 100	0C
4	← ↑ ↑	11 100 100	E4
5	↑ ← ←	10 111 111	BF
6	↓ ↓	00 110 110	36
7		00 000 000	00

汉字“布”的字形点阵如图 7-3-9 所示，绘图向量图示于图 7-3-10 中。

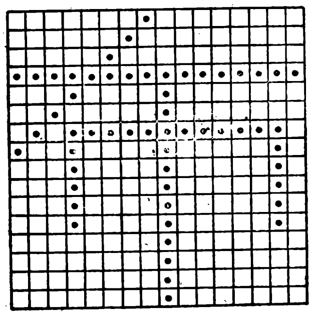


图 7-3-9

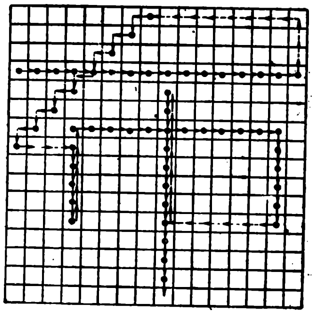


图 7-3-10

绘图向量在内存中的存贮形式如下：

字节	绘图向量			内存存贮形式			十六进制值
	C	B	A	C	B	A	
0		→	→	00	101	101	2D
1		→	→	00	101	101	2D
2		→	→	00	101	101	2D
3		→	→	00	101	101	2D
4		→	→	00	101	101	2D
5		→	→	00	101	101	2D
6		→	→	00	101	101	2D
7		↑	→	00	100	100	25
8	←	↑	↑	11	000	000	C0
9	←	←	←	1	1011	011	DB
10	←	←	←	11	011	011	DB
11	↓	←	←	10	111	011	BB
12		↓	←	00	010	111	17
13		↓	←	00	010	111	17
14		↓	←	00	010	111	17
15		↓	←	00	010	111	17
16		↓	←	00	010	111	17
17		↓	←	00	010	111	17
18	↓	↓	→	01	001	101	4D

19		↓	↓	00	110	110	36
20		↓	↓	00	110	110	36
21		↑	↑	00	100	100	24
22		↑	↑	00	100	100	24
23		→	↑	00	101	100	2C
24		→	→	00	101	101	2D
25		→	→	00	101	101	2D
26		→	→	00	101	101	2D
27		→	→	00	101	101	2D
28		→	→	00	101	101	2D
29		↓	↓	00	110	110	36
30		↓	↓	00	110	110	36
31	←	←	↓	11	111	110	FE
32	←	←	←	11	011	011	DB
33		↑	←	00	100	011	23
34		↑	↑	00	100	100	24
35		↑	↑	00	100	100	24
36		↑	↑	00	100	100	24
37	↓	↓	↓	10	010	110	96
38	↓	↓	↓	10	010	010	92
39		↓	↓	00	110	010	32
40		↓	↓	00	110	110	36
41		↓	↓	00	110	110	36
42				00	000	000	00

其它字形的绘图向量不再一一例举。

假设这个图形表的起始地址为 $6000，这个图形表占用 $6000～$6130 的存贮区域。其二进制数据如下所示：

```
6000-10  00  30  00  37  00  3F  00
6008-49  00  52  00  5C  00  67  00
6010-6F  00  79  00  83  00  8C  00
6018-8F  00  BA  00  C9  00  F1  00
6020-20  01  00  00  00  00  00  00
6028-00  00  00  00  00  00  00  00
6030-3B  2F  24  24  BC  06  00  3F
6038-67  0C  0C  E4  BF  36  00  20
6040-FC  0C  05  38  3F  96  72  2D
6048-00  23  2C  3F  27  0C  0C  36
6050-06  00  3B  07  68  21  1C  3F
6058-24  2D  2D  00  3B  07  20  2C
6060-1C  0C  2D  D6  0E  36  00  1B
6068-24  0C  05  20  3F  3F  00  20
6070-1C  0C  1C  BF  0E  1E  76  2D
```

```
6078-00  3B  27  08  1C  64  AD  36
6080-77  06  00  3B  07  20  24  0C
6088-AD  36  36  00  1B  3B  00  2D
6090-2D  2D  2D  2D  2D  2D  25  C0
6098-DB  BB  BB  17  17  17  17  17
60A0-17  4D  36  36  24  24  2C  2D
60A8-2D  2D  2D  2D  36  36  FE  DB
60B0-23  24  24  24  96  92  32  36
60B8-36  00  41  48  08  17  17  17
60C0-36  36  36  36  AE  15  15  05
60C8-00  12  0C  0C  0C  0C  0C  16
60D0-2D  2D  2D  2D  2D  1E  1E  1E
60D8-1E  1E  1E  1E  1E  1E  36  2E
60E0-2D  2D  2D  2D  24  96  DB  DB
60E8-DB  DB  24  24  24  24  24  24
60F0-00  92  2D  2D  2D  2D  2D  2D
60F8-2D  92  92  E2  1C  1C  1C  1C
6100-1C  E7  E4  1C  1C  1C  4C  49
6108-36  36  36  36  36  36  36  E6
6110-D8  DB  63  0C  0C  0C  0C  0C
6118-0C  0C  0C  0C  0C  0C  0C  00
6120-49  49  41  08  08  15  15  15
6128-36  36  36  36  BE  17  17  07
6130-00  FF  00  00  FF  FF  00  00
```

共占 305 个字节。

图形表的二进制数据输入内存之后，用如下的 DOS 命令保存到磁盘上：

BSAVE CHAR，A$6000，L305

例 19 解放以来，我国轻工产品产量大幅度增长。其中布匹增长情况如图 7-3-11 所示。

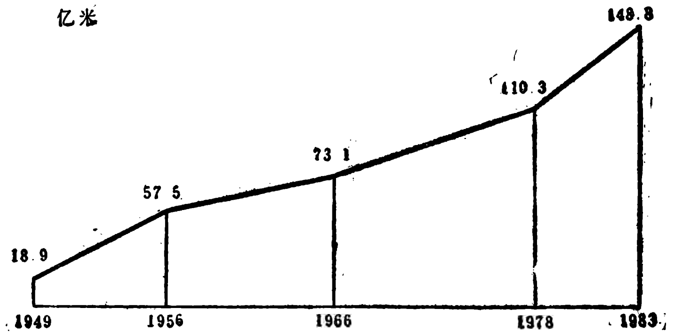


图 7-3-11

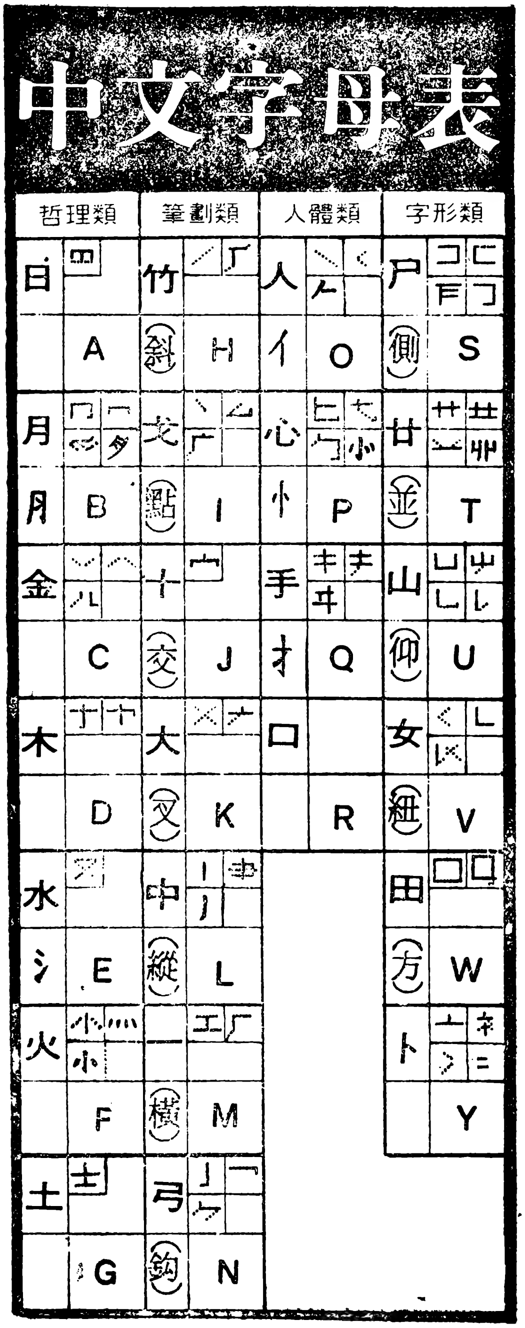
试用图形表的绘图方式绘出这个图表并配注说明。

图形表已在例 18 中建立，调用它给图表配注说明的程序如下：

10 PRINT CHR$(4)；"BLOAD CHAR"　　　　装入图形表

```
20 POKE 232, 0:POKE 233, 96
30 HGR2:HCOLOR=6
40 Y(1)=18.9:Y(2)=57.7:Y(3)=73.1:Y(4)=110.3:Y(5)      给出产量
   =148.8:Y =160
50 YE(1)=1949:YE(2)=1956:YE(3)=1966:YE(4)=1978:YE(5)=1983 给出年份
60 X(1)=38                                          计算各个年份的水平坐标值
70 FOR I=1 TO 4
80 X(2)=X(1)+(YE(I+1)-YE(I))*6
90 HPLOT X(1), Y TO X(1), INT(Y-Y(I)) TO X(2), INT(Y-Y       绘图
  (I+1)) TO X(2), Y
100 X(1)=X(2)
110 NEXT I
120 HCOLOR=2
130 HPLOT 0,Y TO 279, Y                                  绘水平轴
135U$="0"
140 X=48:Y=167:N=YE(1):GOSUB 500                         配注年份
150 X=90:N=YE(2):GOSUB 500
160 X=150:N=YE(3):GOSUB 500
170 X=222:N=YE(4):GOSUB 500
180 X=252:N=YE(5):GOSUB 500
190 X=48:Y=INT(155-Y(1)):N=18.9:GOSUB 500                配注产量
200 X=90:Y=INT(155-Y(2)):N=57.7:GOSUB 500
210 X=150:Y=INT(155-Y(3)):N=73.1:GOSUB 500
220 X=222:Y=INT(155-Y(4)):N=110.3:GOSUB 500
230 X=252:Y=INT(155-Y(5)):N=148.8:GOSUB 500               配汉字
240 FOR U=1 TO 5
250 X=38+(U-1)*16:Y=30:U$=STR$(U):GOSUB 500
260 NEXT U
270 END
500 N$=STR$(N)                                          调用图形表子程序
510 HCOLOR=3:SCALE=1:ROT=0
515 L=L EN(N$)
520 FOR J=L TO 1 STEP-1
530 M$=MID$(N$, J, 1)
540 IF M$="·"THEN DRAW 11 AT X, Y:GOTO 590
550 IF M$="0"THEN DRAW 10 AT X, Y:GOTO 590
560 IF U$<>"0" THEN 620
570 M=VAL(M$)
580 DRAW M AT X, Y
590 X=X-5
```

```
600 NEXT
610 RETURN
620 DRAW U+11 AT X, Y
630 X=X+16, RETURN
```


中文字母表
哲理類
筆劃類
人體類
字形類
日 A
月 B
金 C
木 D
水 氵 E
火 F
土 G
竹 斜 H
戈 點 I
十 交 J
大 叉 K
中 縱 L
一 橫 M
弓 鈎 N
人 亻 O
心 忄 P
手 扌 Q
口 R
尸 側 S
廿 並 T
山 仰 U
女 紐 V
田 方 W
卜 Y

附录 II APPLE SOFT BASIC 保留字

ABS	AND	ASC	AT	ATN	CALL
CHR$	CLEAR	COLOR=	CONT	COS	DATA
DEF	DEL	DIM	END	EXP	PLASH
FN	FOR	FRE	GET	GOSUB	GOTO
GR	HCOLOR=	HGR	HGR2	HIMEM:	HLIN
HOME	HPLOT	HTAB	IF	IN#	INPUT
INT	INVERSE	LEFT$	LEN	LET	LIST
LOAD	LOG	LOMEM:	MID$	NEW	NEXT
NORMAL	NOT	NOTRACE	ON	ONERR	OR
PDL	PEEK	PLOT	POKE	POP	POS
PRINT	PR#	READ	RECALL	REM	RESTORE
RESUME	RETURE	RIGHT$	RND	ROT=	RUN
SAVE	SCALE=	SCRN(	SGN	SHLOAD	SIN
SPC(	SPEED=	SQR	STEP	STOP	STORE
STR$	TAB(	TAN	TEXT	THEN	TO
TRACE	USR	VAL	VLIN	VTAB	WAIT
XDRAW					

附录 III　ASCII 码表

十进制	十六进制	字元	键　入
0	00	NULL	ctrl @
1	01	SOH	ctrl A
2	02	ETX	ctrl B
3	03	ETX	ctrl C
4	04	ET	ctrl D
5	05	ENQ	ctrl E
6	06	ACK	ctrl F
7	07	BEL	ctrl G
8	08	BS	ctrl H or <-
9	09	HT	ctr I
10	0A	LF	ctrl J
11	0B	VT	ctrl K
12	0C	FF	ctrl L
13	0D	CR	ctrl M or RETURN
14	0E	SO	ctrl N
15	0F	SI	ctrl O
16	10	DLE	ctrl P
17	11	DC1	ctrl Q
18	12	DC2	ctrl R
19	13	DC3	ctrl S
20	14	DC4	ctrl T
21	15	NAK	ctrl U or->
22	16	SYN	ctrl V
23	17	ETB	ctrl W
24	18	CAN	ctrl X
25	19	EM	ctrl Y
26	1A	SUB	ctrl Z
27	1B	ESCAPE	ESC
28	1C	FS	n/a
29	1D	GS	ctn shift-M
30	1E	RS	ctrl
31	1F	US	n/a

十进制	十六进制	字元	键入
32	20	SPACE	space
33	21	!	!
34	22	"	"
35	23	#	#
36	24	$	$
37	25	%	%
38	26	&	&
39	27	'	'
40	28	(	(
41	29	)	)
42	2A	*	*
43	2B	+	+
44	2C	,	,
45	2D	-	-
46	2E	.	.
47	2F	/	/
48	30	0	0
49	31	1	1
50	32	2	2
51	33	3	3
52	34	4	4
53	35	5	5
54	36	6	6
55	37	7	7
56	38	8	8
57	39	9	9
58	3A	:	:
59	3B	;	;
60	3C	<	<
61	3D	=	=
62	3E	>	>
63	3F	?	?

十进制	十六进制	字元	键入
64	40	@	@
65	41	A	A
66	42	B	B
67	43	C	C
68	44	D	D
69	45	E	E
70	46	F	F
71	47	G	G
72	48	H	H
73	49	I	I
74	4A	J	J
75	4B	K	K
76	4C	L	L
77	4D	M	M
78	4E	N	N
79	4F	O	O
80	50	P	P
81	51	Q	Q
82	52	R	R
83	53	S	S
84	54	T	T
85	55	U	U
86	56	V	V
87	57	W	W
88	58	X	X
89	59	Y	Y
90	5A	Z	Z
91	5B	[	n/a
92	5C	/	n/a
93	5D	]	] (shift-M)
94	5E	^	^
95	5F	—	n/a

附录IV　中文字母的 ASCII 码表

中文字母	十进制 ASCII 码	十六进制 ASCII 码	英文形码
日	97	61	A
月	98	62	B
金	99	63	C
木	100	64	D
水	101	65	E
火	102	66	F
土	103	67	G
竹	104	68	H
戈	105	69	I
十	106	6A	J
大	107	6B	K
中	108	6C	L
一	109	6D	M
弓	110	6E	N
人	111	6F	O
心	112	70	P
手	113	71	Q
口	114	72	R
尸	115	73	S
廿	116	74	T
山	117	75	U
女	118	76	V
田	119	77	W
卜	121	79	Y

附录 V APPLE SOFT BASIC 出错信息

? BAD SUBSCRIPT ERROR

数组的下标值不正确或下标值超过了定义的范围，错误代码为 107。

? CAN'T CONTINUE ERROR

在下列三种情况下(i)没有程序；(ii) 在有严重错误出现；(iii) 在修改过程序以后，试图使用 CONT 指令继续执行程序。

? DIVISION BY ZERO ERROR

用值为 0 的算式做除数，错误代码为 133。

? FORMULA TOO COMPLEX ERROR

在一个行号中写入两条以上的 IF……THEN 语句，错误代码为 191。

? ILLEGAL DIRECT ERROR

在直接式中执行 INPUT，DEFFN 或 GET 语句。

? ILLEGAL QUANTITY ERROR

在一个字符串函数、数值函数等语句中，所用到的数值超过了能够接受的范围，错误代码为 53。

?NEXT WITHOUT FOR ERROR

执行一句没有 FOR 语句与之对应的 NEXT 语句，错误代码为 0。

? OUT OF DATA ERROR

在输入时，DATA 语句中的数据已用完，READ 语句仍试图读数据，错误代码 42。

? OUT OF MEMORY ERROR

程序太大，变量或数组太多，FOR……NEXT 循环语句嵌套超过 10 层，子程序层次超过 24 层，括号的深度超过 36 层，LOMEM:定得太高，HIMEM:定得太低等引起的错误，错误代码 77。

? OVERFLOW ERROR

从键盘上输入的数或程序计算的结果溢出(机器能接受的范围是 -1.7E+38 与 1.7E+38 之间)，错误代码 69。

? REDIM'D ARRAY ERROR

二次定义了数组空间，错误代码为 12。

? RETURN WITHOUT GOSUB ERROR

执行到一个没有 GOSUB 与之对应的 RETURN 语句，错误代码为 22。

? STRING TOO LONG ERROR

定义一个字符串或多个字符串连接成一个字符串时长度超过了 255 个字符，错误代码为 176。

? SYNTAX ERROR

语句中出现语法错误以及任何不能由其它信息所提示的错误都产生这个出错信息，错误代码为 16。

? TYPE MISMATCH ERROR

使用的变量或数据类型与要求的不符合，错误代码为 163。

? UNDEF'D FUNCTION ERROR

使用了一个未定义过的函数，错误代码为 224。

? UNDEF'D STATEMENT ERROR

程序中试图转向一个不存在的语句编号，错误代码为 90。

附录 VI DOS 错误信息

DISK FULL

软盘中提供给用户使用的496个扇区已经被写满，用户还试图写入程序或数据，错误代码为9。

END OF DATA

文本文件中的数据已经取完，仍试图读数据，错误代码为5。

FILE LOCKED

使用 SAVE, BSAVE, WRITE, DELETE 或 RENAME 对一个已被封锁的文件进行操作，错误代码为10。

FILE NOT FOUND

在磁盘目录表中找不到 DOS 命令要操作的文件，错误代码为6。

FILE TYPE MISMATCH

使用的 DOS 命令和所操作的文件类型之间配合不当(LOAD, RUN, SAVE 命令只能用于APPLE SOFT BASIC 程序文件，OPEN, READ, WRITE, APPEND, POSITION, EXEC 命令只能用于文本文件，BLOAD, BSAVE, BRUN 只能用于机器语言的程序文件)，错误代码为13。

I/O ERROR

在磁盘上读写不成功。通常是磁盘驱动器的门没关上；磁盘没有被初始化过；磁盘驱动器中没有磁盘片或软盘损坏等，错误代码为8。

LANGUAGE NOT AVAILABLE

当使用 FP 或 INT 命令改变目前系统中使用的语言，而想要使用的语言却又不在 ROM 或磁盘中；写入或执行目前系统不允许使用的语言程序则产生这个错误，错误代码为1。

NO BUFFERS AVAILABLE

无缓冲区可被利用，错误代码为12。

NOT DIRECT COMMAND

在语句中直接使用 OPEN, READ, WRITE, APPEND, POSITION 命令(这些语句只能用在 PRINT 语句中)，错误代码为15。

PROGRAM TOO LARGE

用 DOS 命令把程序文件装入到内存，内存容量不够，错误代码为14。

RANGE ERROR

DOS 命令中某参数超出了规定的范围，错误代码为2或3。

SYNTAX ERROR

DOS 命令中有语法错误，如标点用错，参数次序不对等，错误代码11。

VOLUME MISMATCH

DOS 命令中V参数的值与磁盘上记录的卷号不符，错误代码为7。

WRITE PROTECTED

试图在一个具有写保护的磁盘上使用 SAVE, BSAVE 或者 WRITE 命令，错误代码为4。

科技新书目：90—200
统 一 书 号：15335·011
定　　　价：2.05元